AUTOMATED CHECKING OF BUILDING REQUIREMENTS ON CIRCULATION OVER A RANGE OF DESIGN PHASES

A Dissertation
Presented to
The Academic Faculty

by

Jae Min Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Architecture

Georgia Institute of Technology
August, 2010

AUTOMATED CHECKING OF BUILDING REQUIREMENTS ON CIRCULATION OVER A RANGE OF DESIGN PHASES

Approved by:

Charles M. Eastman, Advisor
College of Architecture and Computer
Georgia Institute of Technology

Dr. John Peponis
College of Architecture
Georgia Institute of Technology

Dr. Ellen Do
College of Architecture and Computer
Georgia Institute of Technology

Dr. Calvin Kam
School of Engineering
Stanford University, GSA

Wawan Solihin
Dep. of Software Development
Autodesk

Date Approved: July, 02, 2010

"If anyone deserves the title "father of BIM", surely it is Chuck Eastman..."

Jerry Laiserin, BIM Handbook 2008.

ACKNOWLEDGEMENTS

This dissertation could be written thanks to many support and feedback of many people. First and foremost, I especially want to thank my advisor, Prof. Chuck Eastman, for his guidance during my research and Ph.D. study at Georgia Tech. His perpetual energy and enthusiasm in research had motivated all his advisees, including me. Prof. Eastman has been my inspiration as I hurdle all the obstacles in the completion this research work.

Prof. John Peponis, Prof. Ellen Do, Dr. Calvin Kam, and Manager Wawan Solihin deserve special thanks as my thesis committee members and advisors. In particular, I would like to thank Prof, John Peponis, Head of Ph.D. Program, for the insights he has shared for my thesis, Prof. Ellen Do for her encouragement for Ph.D. Study, Dr. Calvin, a Senior Program Expert of National 3D-4D-BIM Program in GSA, for his advice on GSA courthouse automation project, and Wawan Solihin, a manager of Dep. of Software development in Autodesk, for reviewing and giving precious comments on my thesis. This thesis could be written by helps from GSA and Solibri because this thesis was motivated from GSA-U.S. Courts Design Guide Automation project, and theory in my thesis was implemented by using Solibri system. I appreciate Peggy Ho Yee and Fred miller in GSA, and Pasi Passiala in Solibri.

I would like to give special thank to Prof. Uk kim, hongik Univ., my advisor in master program, Prof. Jinwon Choi, Yonsei Univ.,and Prof. Sungah Kim, Sungkyunkwan Univ. Thank to their teaching and recommendation, I could have a chance to study for Ph.D. in Georgia Tech, and finally I could finish it.

There are several people, who left Georgia Tech, but I owe acknowledgements while I have been studying in Ph.D. program. Prof. Ghang Lee, Yonsei Univ., Prof. Hyeonjoon Moon, Dankook Univ., Prof. Soonyook Kwan, Sungkyunkwan Univ., and Prof. Cheolsoo Park, Sungkyunkwan Univ., showed me a path to follow for successful Ph.D study.

Thank to my friends in Georgia Tech, I could enjoy the life in Ph.D study. Thank my friends; Yeon suk Jeong, Junha Kim, Donghoon Yang, Frank Wang, SukJoon Yoo, Jinkook lee, Sherif Abdelmohsen, Hugo Sheward, Paola Sanguinetti , Elif Yagmur, Sanghoon Lee, Huafen Hu, and Sunhay Kim.

I would like to thank my friends in korea; MD. Inmo Jung, MD. Honggun Park, Seunyong Park, Yeongseop Shin, Sangsoo Lee, Giyoung Jang, Ilyoong Kwak, Hyunbum Choi and Tahoo Kim. They all treated me with real friendship whenever I visit korea.

Last but not the least, my deepest gratitude goes to my family for their unflagging love and support throughout my life; this dissertation is simply impossible without them. I am indebted to my father,Gibum Lee, and mother, Mansuk Whang, father in law, Chaesoon Park, mother in law, JungJa Kim, brothers, Jaeman Lee, Jaebok Lee, Jaecheon Lee, sister Jaewon Lee, brother in law Jeongsik Yang, Hyunjoo Park and sister in law, Giyien Park for their care and love. As a typical Korean family, they worked industriously to support the family and spare no effort to provide the best possible environment for me to attend school.

I cannot ask for more from my wife, Nana Park, as she is simply perfect. I have no suitable word that can fully describe her everlasting love to me. She had never complained in spite of all the hardships in her life to support me. Finally I thank my daughter and son; Seoyeon (Annette) Lee and Chanhee (brandon) Lee. Whenever I have a difficulty in my life, they remain me how beautiful the life is and give a reason to live.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Automated building rule checking is an automated process of design evaluation against design requirements. Since the early 1970's, when the electronic representation of building design became available, automated building rule checking, a computational process, has been a focus of study, and it continues to be a popular research area because it facilitates the design evaluation process by reducing the checking time and evaluation costs and by increasing the objectivity and the reliability of the evaluation. Thanks to the emergence of BIM (Building Information Model) authoring software, BIM became available to use in real building design, and several automated building code checking systems are being developed based on BIM.

In practice, the use of a rule checking system in real design evaluation may be influenced by several factors. Among the factors that affect the accuracy and the reliability of automated checking such as checking algorithms and rule interpretation is the level of completeness of the BIM in the design process, which can cause limitations in the application of a rule checking algorithm to the model. Problems caused by the incompleteness of the building model occurred in CORENET project, a project initiated by the Singapore government in 1999 for automation of building code checking, and GSA U.S. Courts Design Guide Automation project (GSA Project), initiated at Georgia Tech in 2007 also faced with the same problems caused by incompleteness of building model in the development stage.

This thesis is a continuation of the research of GSA-U.S. Courts Design Guide Automation project (Simply, GSA project). This study focuses on the development of a new, formal method for the automated checking of occupant circulation rules in *U.S. Courts Design Guide*. The theoretical goals of this study are to provide a logical foundation upon which one can build an automated checking module for circulation rule checking and that is capable of outlining the rule-validation process independently from

its diverse implementation. The theory for circulation rule checking is devised to represent the process of the validation of a building design in the development stage. The theory deals with issues of validation caused by the lack of data in the development of a building design.

As a preliminary research to figure out the main issues in the development of a generic circulation rule checking module the research started from detailed review of occupant circulation rules in *U.S. Courts Design Guide*. Based on the review of the occupant circulation rules, this study classifies the rules by types according to the common properties of occupant circulation rules and then defines the scope of occupant circulation rules. It also defines the conceptual structure of the rules within this scope to create a framework within which consistent rules can be interpreted; and based on the conceptual structure of the rules, it develops a computable representation of occupant circulation using parameters intended to define the conditions of the rules.

As a part of GSA project, the implementation of the framework begins with the development of the automatic checking module only for specific design stages (Preliminary concept design stage and Final concept design (FCD) stage), which are defined in Public Building Service (PBS) Design Guide[1] . A typical model in the preliminary concept design, comprised of aggregated spaces not yet developed into individual spaces, and the model has no doors that explicitly connect spaces. The final concept design, containing individual spaces, has doors that explicitly represent the concept of a space connection. and physical or virtual wall bounding spaces .

The GSA project team developed two individual methods for checking a design during the preliminary concept design and final concept design stages. The method for the final

[1] Design guide for public building owned by the civilian federal government. It is available at GSA website : http://www.gsa.gov/Portal/gsa/ep/contentView.do?contentType=GSA_OVERVIEW&contentId=8062&noc=T

concept design is based on connections between spaces through elements such as doors, openings, elevators, and stairs. These connections are represented to a graph named as a topological connection graph, simply a connection graph used to conduct circulation checking. The checking method for the preliminary concept design is implemented based on aggregated adjacency spaces with the same required property of the rule because preliminary concept design does not contain explicit connections from space to space. Checking is performed based on the assumption that if the spaces in an adjacent set possess the same required property, then they are potentially connected to the required property between them. This method generates multiple sets of adjacent spaces, and rules are checked based on the sets.

The method of checking for the final concept design is referred to as the topological connection graph-based method (briefly, graph-based method), and the method for the preliminary concept design is referred to as set-based method. These methods are applied to the validation of multiple real courthouses in final concept design and preliminary concept designs, and the application of these methods successfully identified real design errors in less than a minute.

And the checking results were reported to GSA for review.

However, the research performed for GSA Project reveal a following issue in automated checking of Courthouse models made by architects.

- ***Limitation in the coverage of the model in the development stage***

The graph-based method can check a courthouse model in the final concept design against all conditions outlined in the occupant circulation rules from U.S. Courts Design Guide. However, this method is only applicable in cases in which the graph can be generated from a model with information about space connectivity. Thus, this method cannot check the model without connection elements such as doors, openings, or stairs.

Even though the set-based checking method can check a model in the preliminary concept design stage, the checking algorithm totally depends on the sets, which are an aggregation of adjacent spaces with the same properties. Thus, the application of set based checking method is limited only for a model in preliminary concept design.

In order to check a model in between preliminary concept design and final concept design, the model should be simplified to a preliminary concept design model or modified to a final concept design model by anyone who checks this model. This simplification causes a loss of existing data in the model, which might cause inaccurate checking results. And modification to afinal concept design model will lead to changes in the design by a third party, which will in turn lead to inaccurate checking results also.

Another consequence of simplification and manual modification is that the preparation of a model for checking may become more time-consuming. Although the time required for checking takes less than a minute using the automated checking methods, preparing the model for checking takes several days because a person must be careful not to change the original design intent. In some cases, the required design change that needs to be checked was serious, so the model could not be checked until the architect had finished the modification.

To overcome the limitations in the modules developed in the initial investigation, this thesis focuses on the generalization of checking to cover models in all design stages including preliminary concept design and final concept design**.** To achieve these goals, this thesis focuses on the following issues.

- ***Research on the diversity of building design in the development stage***

Building design in the development stage can vary from the diverse perspective of completeness. It can have diversity in completeness at the building entity level, such as whether it has doors, walls, aggregated space, and so forth. It also varies according to the completeness of department levels, floor level, and building system level, such as the MEP system. From an almost immeasurable amount of diversity, the cases meaningful for occupant circulation rule checking have been investigated. The strategy to deal with lack of information in each case has also been studied.

- ***Logical representation of checking algorithms***

One of the issues for the generalization of the checking method is to develop a generic algorithm flexible enough to cover the diversity of models in the development stage. The algorithm should also be independent of the diverse way of modelling and implementation. To achieve this goal, the checking process is described logically based on the abstracted representation of requirements in the rules and the building models in the development stage. The checking algorithm is developed based on the logical representation of the checking process. With this approach, the checking algorithm is independent from the implementation method of a building model and programming languages.

For testing and verifying the developed algorithm, this thesis also implements the algorithm using the following methods.

- ***Graph-based representation of space connectivity and space adjacency***

For representing routes in a building, this thesis adopts two types of graphs: the connectivity graph and the adjacency graph. The connectivity graph is intended to represent currently existing routes in a building, and the adjacency graph is designed to capture potential connections between spaces according to the design developments. By using these two graphs, the thesis validates a model not only as it is but also as it could be.

- ***Object-oriented representation of building model and rule requirements***

In order to have more flexibility of implementation in terms of the extension of the type of building model and rules for circulation checking, object oriented representation of building model and circulation rules are developed. Even the main rule for circulation checking method is the rules in U.S. Courts Design Guide, the object oriented classes structure of checking method is developed from the logical process of circulation rule checking so that the checking method can be extensible to cover other type of rules in future needs without changing the existing structure.

The implemented system is verified on two levels. First, the system checks test models intended to show all possible cases in a development model. Second, the system checks real courthouse models in the development stage. In addition, it checks whether the system can accurately detect all errors in the model by validating all issues the system found.

The contribution of this thesis is that it is the first study pertaining to the logical representation of the automated checking of circulation rules considering design development. The logical representation of checking process is used to develop an object-oriented class structure of the rule checking system with the intention of extending it to cover other types of rules by defining subclasses of the existing classes without changing the main structure.

By the development of automated circulation checking module and application of it to real courthouse validation, the module can find real design issues in couple of minutes from a courthouse model in developments. The reporting module embedded in the checking module can visualize problematic spaces, which cause violation of circulation requirements. And this visualization allows to designer to see the issues in design intuitively.

CHAPTER 1

INTRODUCTION

1.1 Purpose

This thesis aims to develop a new method for automatically checking occupant circulation rules for courthouse design. Specifically, the purpose of the new method is to check a courthouse design during all stages of the development process from the preliminary concept design to the final concept design.

1.2 Motivation

The nature of the building design process is the continuous repetition of a design creation and review rather than just one seamless progression. Typically, an architect does not develop a design without repetitive revisions because the design itself rarely progresses without an architect encountering problems that call for revisions, and revisions typically cause further design problems. Most of design problems are associated with other problems. Thus, resolving of a design problem could cause another problems rather than resolving of an isolated problem.

From the perspective of problem solving, a design is a process of problem solving by iterative reviews of a previous design. The checking of a building design against design requirements is a type of design review; thus, design checking also is an iterative process throughout the development of a design. Checking a design within the context of design development is critical because it allows an architect to identify and resolve design problems, which in turn leads to better design development. Thus, checking of design against design requirement is performed manually by architect along with design development.

Several studies on the automatic checking of design requirements have attempted to identify ways to overcome the limitations of manual checking such as subjective and non-thorough checking. They have focused on several issues such as the translation of design requirements to computational form [1,2,3], the abstract representation of building design with graphs or object-oriented models [7,9,11], and diverse rule-checking algorithms [5,12,]. Multiple commercial systems such as E-Plan Checker [39] and Solibri system [33] have also been developed or are still under development [see Section 2.3 for more details].

Despite the numerous studies and systems pertaining to the automatic checking of design requirements, most do not deal with building design in the development stage but instead they are limited to the checking of a building design during a specific phase such as the final design for submission to authorities for approval rather than all the design stages during the development of the design.

If a model can be checked during the entirety of the development phase, the checking process would be more practical as designer could identify design errors instantly after making design changes, thus allowing the designer to develop a more practicable design with more concrete information about the problems caused by previous changes. Current research and rule checking systems limited to a specific stage of design do not support the immediate and continuous feedback of design changes in design development.

The motivation for this thesis stems from the limitations of current rule checking systems, restricted to nearly completed building designs only.

1.3 Content of the thesis

As research intended to create a more practical rule checking system, this thesis investigates possible algorithms for checking occupant circulation rules automatically based on a building model in various design stages and suggests new methods of approaching the problem. As a real instance of occupant circulation rules, real rules outlined in the *U.S. Courts Design Guide* were analyzed, and initial rule checking methods are developed for checking the courthouse rules. Based on the experience of developing the initial rule checking methods, this research develops a generic checking method.

Occupant circulation rules are selected because they should be considered in both the early stage of the design for the space layout to the late stage of the design for details of spaces and their connections. To clarify the scope of this thesis more clearly, occupant circulation rules are defined [see Section 3.2.3].

In the preliminary development of a checking method, parameters for interpreting circulation rules are devised, and as a checking method, the *graph-based method* and the *set-based method* are developed and tested.   The *graph-based method* is developed to check the circulation rules as they apply to the final concept design and the *set-based method* is developed to check the rules as they apply to the preliminary concept design. However, these methods are applicable only to the specific stage of the final concept design and preliminary concept design models. Based on the results of the preliminary research, a more general method that can check models in all design stages from preliminary concept design to final concept design is investigated.

This thesis consists of eight chapters, the contents of which are summarized below.

- **CHAPTER 2 - Background**

This chapter explains how this research was initiated and reviews previous research on occupant circulation rule checking. It reviews studies on the interpretation of rules to be computable for automatic checking and describes research on diverse methods for checking such as simulation- and graph-based methods.

- **CHAPTER 3 - Analysis of Occupant Circulation Rules**

This section describes what occupant circulation rules are, how the occupant circulation rule can be classified, which occupant circulation rules fall within the scope of this thesis, and how the rules can be interpreted for computation. In particular, this chapter identifies parameters that define the requirements in the rules and shows how the occupant circulation rule can be interpreted by using the parameters as a means of computation.

- **CHAPTER 4 - Graph and Set-Based Checking**

This chapter describes the preliminary methods developed for checking occupant circulation rules in the *U.S. Courts Design Guide* as a part of GSA project It includes two methods: the *graph*- and *set-based methods*, which have been developed for checking a building model at specific design stages; Late and early concept stages. It explains these two states of design and provides details about how the methods have been developed specifically for checking building models during these two concept stages.

- **CHAPTER 5 - Generalization of Checking Method**

This chapter proposes a generic method of overcoming the limitations of the set- and graph-based methods. It contains a detailed discussion of the limitations of these methods and approaches to overcoming these limitations. It includes a logical representation of the occupant circulation rule and its validation process with building design. By using a logical description, this chapter also provides a logical description of the checking process with consideration of design development.

- **CHAPTER 6 - Application of the General Checking Method to Courthouse Design**

This chapter shows how the generic method can be used to check the occupant circulation rules in the *U.S. Courts Design Guide*. It provides a detailed discussion on how this method can check diverse conditions in the rule. It also contains the pseudo code of the generic method for the automated checking of the rules.

- **CHAPTER 7 - Implementation**

This chapter illustrates the modules that comprise the generic checking method and its structure, implemented on top of Solibri Model Checker. It involves an object-oriented class structure that represents the logical description of the generic checking process used for real implementation. It explains the functionality of each module and the process of rule checking via the modules.

- **CHAPTER 8 - Validation**

The implemented generic method is validated by the checking of authentic diverse building models during the development stages.   It will test the results of the generic method through the checking of building test models and eleven authentic courthouse building models. It also describes the four processes of verification of the checking results.

- **CHAPTER 9 - Discussion and Conclusion**

This chapter discusses the principal issues, which are found after applying generic method to real courthouse model validation. It describes analysis results after reviewing the checking results of several courthouses, and it involves the factors that lead to problems in the application of generic method to the validation of real building models. It also summarizes this research and describes its contributions and limitations.

CHAPTER 2

BACKGROUND

2.1 Overview

This section presents an overview of preliminary research that relates to the automated checking of occupant circulation rules. Because this thesis presents the ongoing research of the GSA U.S. Courts Design Guide Automation Project at Georgia Tech, this chapter begins by introducing the project and then discusses the prior literature.

2.2 Background of the current research

The Georgia Tech GSA project team, led by Professor Chuck Eastman, has been conducting research under the auspices of the U.S. Courts Design Guide Automation Project since 2007. In the initial stages of the project, the team studied the automated checking of occupant circulation rules related to the automation of occupant circulation rule checking, focusing on three aspects: the representation of occupant circulation rules in computable form for automated checking, the representation of building design for circulation checking, and algorithms for checking the model prepared for circulation checking against the computerized circulation rules. The representations and algorithms are developed to apply to two specific design stages of the building design process: the preliminary concept design and final concept design stages [see Section 5.3.5].

- **Representation of the occupant circulation rule in a computable form**

The rules of occupant circulation in the *U.S. Courts Design Guide* were defined, selected, and interpreted in computerized form for automated checking [see Section 5.2]. Eleven pre-defined parameters for interpreting the rules in a computable form were determined, and the circulation rules in the *U.S. Courts Design Guide* were interpreted using the parameters [see Section 5.2].

- **Automated checking of courthouse designs in the final concept design stage**

In order to check a courthouse design in final concept design automatically, circulation rules in the *U.S. Courts Design Guide* were extracted and interpreted using the pre-defined parameters [see Section 5.2] devised so that the rules are in a computable form. A courthouse design also was also prepared to support circulation rule checking by using two types of graphic representations: a topological graph representing space connectivity and a metric graph calculating the metric distance of routes. This study then performed the automated checking of real courthouse designs by using the graphs to interpret the design and by checking the graphs against computerized circulation rules. The automated checking system found several design errors in the real courthouse designs that were subsequently reported to GSA for review [see Section 5.3.4].

- **Automated checking of courthouse designs in the preliminary concept design stage**

In the next step, the research team checked the courthouse designs in the preliminary concept design stage because GSA realized that a design review in the final concept design stage was too late for the errors in the design to be rectified. The designs in the final concept design stage had already been developed in detail, so they hoped to review the courthouse designs in the preliminary concept design stage to avoid the considerable effort required to fix the design in the final concept design stage. However, the automated checking of the preliminary concept design was a challenge because of the lack of information in the preliminary concept design stage. For instance, the preliminary concept design does not have information about the doors or openings that connect spaces for the movement of those inside. Thus, the circulation of people through spaces cannot be calculated explicitly.

In order to check the circulation rules of the preliminary concept design, the set-based method, which uses the potential for space connectivity between groups of spaces instead of explicit space connections, was developed. Potential space connectivity indicates a potential connection between two spaces if they are in a group, generated by the adjacency of each space in the group. Two spaces in different groups indicate no potential route. This assumption is useful when one is checking the type of occupant circulation rules that require accessibility between two spaces in a group such as a “restricted zone.” The team developed a space grouping algorithm based on adjacency and common properties in spaces, and applied them to specific types of rule checking. The real courthouse design (i.e., the courthouse 5 design) was tested during the preliminary concept design stage, and the checking results were reported to the architect for review.

2.3 Research on circulation rule checking

Occupant circulation rules are written in natural language. In order to check the rules automatically with a computer, the rules should be translated to a computable form, which involves two steps: interpreting the rules with quantifiable parameters and mapping the parameters to properties in the building model that need to be checked. First, the rule should be interpreted into quantifiable parameters that can be measured. For example, vague terms such as “long” and “close” should be converted into measurable factors such as “more than 10m” and “less than 10m.” However, more subjective terms such as “efficient” are more difficult to convert into measurable forms.

Several studies have identified quantifiable parameters associated with occupant circulation, particularly that which takes place in airports terminals, hospitals, and shopping malls, where the movement of people is closely related to the functions of the building. These studies have investigated the flow of patients and articles in hospitals [18, 22], the flow of passengers and baggage in airport terminals [19, 20, 31], and the flow of shoppers in malls [21, 24, 26, 27, 28].

The second step in the interpretation process is to map the quantifiable parameters onto properties in the building model. The difficulty of mapping completely depends on the data in a building model. If the data are rich enough to identify all the quantifiable parameters in the rules, the mapping could be straightforward. Otherwise, the mapping requires a process of identification that precedes mapping. When no standard building product model such as IFC (Industry Foundation Class) was available, the identification or the retrieval of required properties from pure geometry data created a critical roadblock. Nguyen [7, 8, 9] studied the retrieval the information pertaining to the connectivity of spaces from a wire frame CAD model as a preliminary step in their research on the automatic checking of fire egress rules. He studied an algorithm for extracting the topological relationships of spaces in a building from the 2D wire-frame CAD model, and Tang-Hung focused on deducing three-dimensional topological relationships among spaces from conventional CAD data.

Thanks to the emergence of standard building product models such as IFC, the mapping process has become more straightforward. In cases in which a building model contains parameters for explicit rule checking, the mapping process involves identifying a parameter from a model that corresponds to another parameter in a rule and then linking them. That is, the checking process is the comparison of these two parameters. Q.Z. Yang [10, 11] defined a java class referred to as the "rule-provision" to map the parameters in a rule and those in a model. He checked the accessibility of a wheelchair through doors by comparing the minimum width for wheelchair access with the width of the doors in the building model. However, fire egress checking requires additional steps because a building model does not explicitly contain egress routes. Several researchers developed algorithms for retrieving routes using relationships among the spaces in a building model. The Singapore-CORENET Project [17] and Kannala [33] developed a module for retrieving routes from the IFC model to meet their own fire egress regulation.

In some cases, even if a rule clearly defines requirements, mapping them to a building model may prove difficult. For instance, even though the rule in the above example describing the accessibility of a wheelchair in a corridor clearly defines the required properties for checking (the minimum width of the corridors), a target object in the building and its properties are still vague because it is hard to define which areas in the corridor should meet the minimum width requirements. Some rules are more ambiguous. Another rule, example2, does not clearly describe the requirements, only the functional requirements of a toilet. Han [4] called these types of rules performance-based rules, which differ from prescriptive rules, which describe requirements according to a set of predefined quantitative properties of an object or multiple objects, defined as normative values after multiple simulations. According to the prescriptive rule, the object in example 1 to be checked is the corridor, and the attribute to be checked is the width of the corridor. In this case, the rule clearly describes an object and its attribute to be checked. However, in the case of performance-based rules, identifying an object and its attribute that must be checked and corresponding elements in a model is a more challenging task.

In one approach to checking a performance-based rule, Han introduced an agent-based simulation approach [Figure 1]. Agent-based simulation is a type of simulation performed by an artificial agent that has pre-defined behaviour patterns associated with a specific purpose. He used a robot movement algorithm to simulate the movement of a wheelchair and then applied the simulation results to accessibility checking.

Rule for Example 1: [Prescriptive rule]

The minimum width of corridors shall be 8 feet (2438 mm) per Section 420A.5.1, and 5 feet (1524 mm) as permitted by Section 420A.5.2.

Rule for Example 2: [Performance-based rule]

A disabled person should be able to access the toilet with a wheelchair.

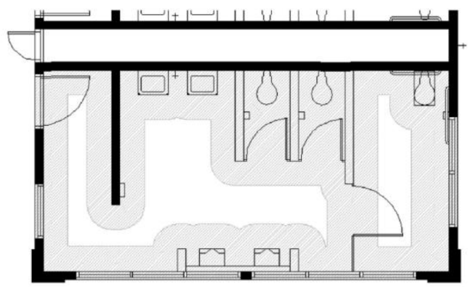


Figure 1 Accessibility checking by agent-based simulation

In Figure 1, the grey area indicates an area that is not accessible to a wheelchair, and the white area is an area that is accessible to a wheelchair.

As an alternative representation of a spatial relationship, graph-based representation has been used in several studies of occupant movement. Because of its well-developed theory and efficient traversing such as Dijkstra's algorithm [45] for finding the shortest path in a weighted, graph provides an excellent performance in graph based checking. Several areas associated with routes such as GIS, robotics, computer games, and space syntax has developed their own graph-based representation to solve their own domain issues. Graph-based representation can be classified into two main types: representation of spatial relationship within a space and that between spaces. This thesis reviews the literature from the perspective of graph representation in both classifications.

First, the computer gaming area has developed several graphs that represent routes within a space. One simple representation of a graph for routes within a space is a regular grid graph with primitive shapes such as a rectangle, a triangle, and a hexagon [Figure2] [35, p. 162]. The nodes of a graph are placed in the points of the primitive shapes and the edges of a graph are placed on the edges of the primitive shapes.

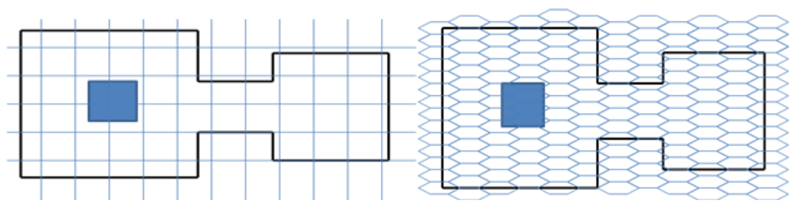


Figure 2 Regular grid-based graphs

Another example of graph-based representation is the corner graph, which uses corner points offset in a space for its nodes. The points are defined from offsets from any corner within the boundary of a space or within the boundary of obstacles in the space [see the first and second graphs in Figure 3].   The edges between the corner points are determined visually from each vertex [34]. All other visible corner points from a corner point are connected. A corner graph is useful for locating a shortest path between two corner points in a space with consideration of obstacles. If the distance of each edge is saved as a weight factor, the corner graph becomes a weighted graph. By using Dijkstra's algorithm, we can easily find the shortest route in a weighted graph. If there is no obstacle in a space, the corner points around the obstacles do not need to be considered. Thus, the graph can be simplified by using convex points in the space [see the second graph in Figure 3]. Any shortest route between two end points (such as doors) in the space should be either through convex points or direct connection between them if the two end points are visible from each other [34].

Another very well-known graph is the circle-based waypoint graph [the third graph in Figure 3]. In this graph, the nodes are placed on the center point of the circles attached with surrounding objects with a maximum radius. This graph is used for checking the accessibility of an agent with a certain clear width [34].

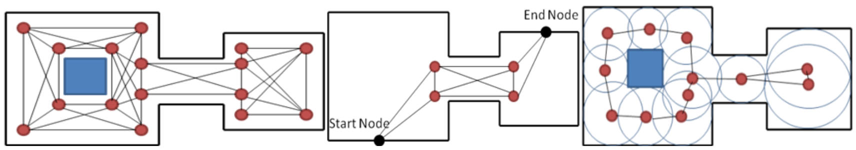


Figure 3 A corner graph and a circle-based waypoint graph

The graph-representation in the gaming applications mostly focuses on the one big open space with obstacles. Thus, even though the regular grid-based approach or corner point-based approach has O^2 efficiency by the increased number of nodes, these approaches are generally used to depict the agent's movement in a space instead of route generation on a whole-building scale [34].

Another type of graph represents the relationship among the spaces in a building. One well-known type of graph that represents spatial relationships is the dual graph. The dual graph of a graph G has a node for each region of G and for each edge in G joining two neighbouring regions [the first graph in Figure 4]. According to its definition of a node and an edge, this graph is conducive to generating a graph for space adjacency from sets of polygons that represent the boundaries of the spaces in a building.

Several researchers have adopted a dual graph concept that generates a graph representing the adjacency of spaces in a building and applied adjacency to the space layout [35, 36, 37].

Another type of graph based on accessibility among spaces is the convex space graph, which represents space connectivity within an entire level of a building [the second graph in Figure 4]. It places a node at the center of the spaces and connects them if they can be connected through openings such as doors or open areas. The edge in this graph represents the connectivity of two spaces, so it indicates that two spaces are accessible by people inside the building.

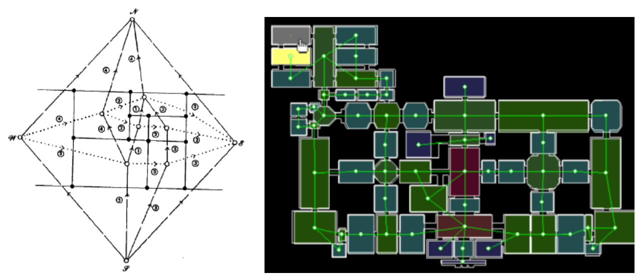


Figure 4 Dual graph and convex space graph

Each graph above is developed for a specific purpose. The graphs in the game area such as the regular grid graph, the corner graph, and the circle-based waypoint graph are mainly developed to depict agent movement within a space with obstacles. Thus, it is beneficial to represent the smooth movement of an agent in a space away from obstacles. However, it focuses on routes within a space instead of connections between spaces. In general, a building consists of many spaces. Thus, these graphs, with too many nodes to apply, represent the spatial relations in a building.

More recent research on circulation checking has been undertaken by the *U.S. Courts Design Guide* Automation Project team, led by Professor Chuck Eastman at Georgia Tech. The team developed two types of graphs: a topological space connection graph (in short, a connection graph) and a metric graph. The connection graph consists of a node for a space and a node for a space-connection element such as a door or an opening.   It represents the connection between spaces through space connection elements. The metric graph is made of a node representing a corner in which turning around occurs. It represents the shortest route between two spaces so that the shortest distance between spaces can be calculated [47]. Chapter 5 describes these graphs in more detail.

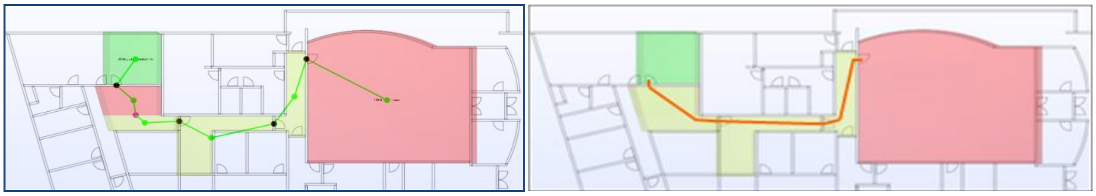


Figure 5 Topological Space Connection Graph (Connection Graph) vs. Metric Graph

2.4 Review of previous research

A simulation-based circulation analysis has been performed on two main approaches to movement: the deterministic approach and the stochastic approach (e.g., Monte-Carlo simulation). The deterministic approach defines agent movement as a combination of determinant variables such as velocity of occupant and movement direction (first image in figure 6). The stochastic approach determines the movement of an agent with an indeterminate variable such as a stochastic variable (e.g., a distribution of agents, second image in figure 6) [46].

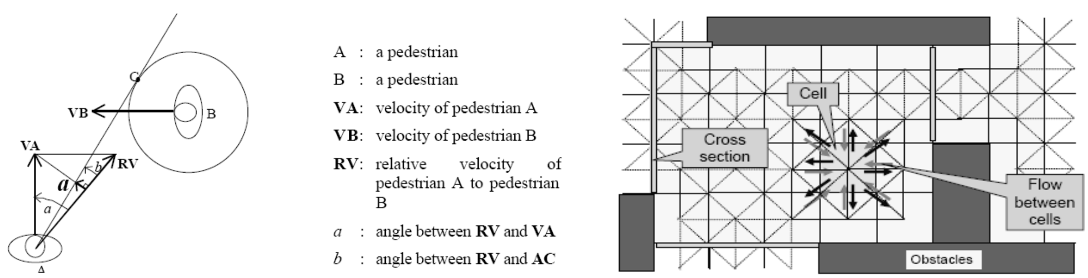


Figure 6 Deterministic approach vs. stochastic approach

Even though checking with agent-based simulation has its own merits such as the realistic movement of agents, it has limitations when it is applied to whole-building scale checking. First, checking by agent-based simulation is costly because it must cover the entire dynamic movement occurring in all the routes. This approach was devised for the movement of one or multiple agents that immediately respond to dynamic changes in environmental conditions and time. Thus, it has been well adopted in computer games and robotics, in which dynamic changes take place frequently. However, a building model is static data, so no dynamic changes occur while rule-checking is being performed. Even if some occupant circulation rules such as congestion in a space are associated with the dynamic movements of person, they can be interpreted as static conditions of the spaces. For instance, a "lobby should be large enough to avoid congestion in peak hours" can be described as a "lobby that should be larger than 1,000 square feet." This thesis focuses on the occupant circulation rules that can be checked by the static properties in the building model.

The graphs for routes within a space such as the corner graph and the circle-based way point graph consist of nodes representing a specific point such as the turning points in a space. These graphs are used to depict a route in a space such as the shortest route between two end doors in a space. The dual graph and the topological connection graph (the connection graph) are graphs representing the relationship between spaces. The dual graph represents the adjacency of spaces, and the connection graph represents the connection of spaces through doors or openings. The connection graph can be used for checking rules on the current connectivity between spaces, and the dual graph can be used for checking rules on potential connectivity.

To deal with the potential of design under development, this thesis adopts space adjacency. Even though several studies conducted by the Georgia Tech team have dealt with occupant circulation rules, current research assumes that a model to be checked contains all the required information needed for checking and mainly focus on how rules can be checked automatically with the assumed model. In the practice of automatic building code checking, a model to be checked could be on a diverse level of completion because it is in the process of development.

The Georgia Tech team has also experienced a number of problems caused by the incompleteness of building models. The team is developing an automated checking method for specific stages in design development: the Preliminary concept design level and the Final concept design level. Each stage has required data that can be checked with specific checking algorithms. However, in practice, most building models the team received for testing from architects were in the development stages between the early and late concepts. These models were not able to be checked directly with one of the developed methods. The team had to spend more time adjusting the model to fit their checking method than performing the checking.

CHAPTER 3

ANALYSIS OF CIRCULATION RULE

3.1 Overview

The preliminary research started with a detailed review of occupant circulation rules in a real design guide, the *U.S. Courts Design Guide*. Based on the review of the occupant circulation rules, this thesis has classified the rules in the guide according to the common properties in occupant circulation rules and defined the scope of the occupant circulation rules for this thesis. This thesis defines the conceptual structure of the rules to develop a framework within which the rules can be consistently interpreted. Based on the conceptual structure of the rule, this thesis develops a computerized representation of occupant circulation according to parameters that are intended to define the conditions of the rules.

As we mentioned above, I am not aware of any research on the automatic checking of diverse occupant circulation rules except rules on fire egress and the accessibility to the disabled has been conducted. This section describes current efforts in research that pertains to the automatic checking of occupant circulation rules in the *U.S. Courts Design Guide*. The description includes an interpretation of the occupant circulation rules for automatic checking and two methods developed for automatic checking: the graph-based method and the set-based method for the GSA project for U.S. Courts.

The following section presents an overview of the automatic checking process as it relates to the *Court Design Guide* [40]. First, the occupant circulation rules had to be extracted from the guide because it has many other rules that do not relate to occupant circulation. The selected rules were then interpreted by parameters so that the rules could be converted into a computerized form. The parameters were devised to represent combinations of diverse circulation rules. The parametric rules represent input to the checking system, which will be ready for use in the checking process. (The rules were loaded to a memory in a rule checking system labeled a "rule set."]

A rule-checking module checked the rule sets using space adjacency, and the results of checking were saved to a report. Figure 7 shows the overall process of automatic occupant circulation rule checking of the *U.S. Courts Design Guide*.

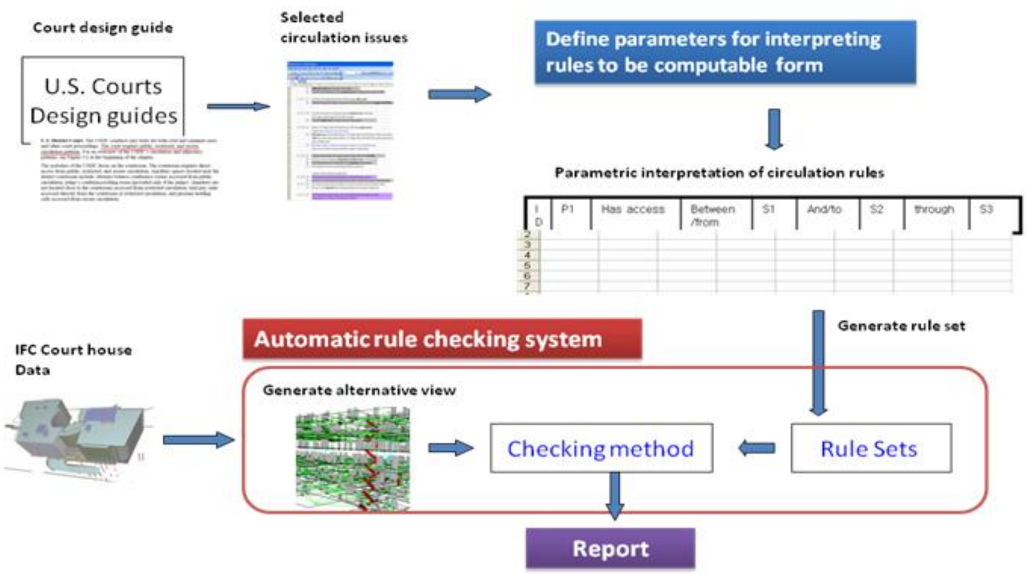


Figure 7 Overview of the automatic rule checking process

3.2 Occupant circulation rules in the U.S. Courts Design Guide

3.2.1 Courthouse design

A courthouse consists of many aggregated and individual spaces with clearly distinct security levels, depicted as spaces in Figure 8. For example, a judge's chamber is an aggregated space comprising a judge's chamber, a toilet, a secretary's office, and a conference room. These spaces should be in a restricted zone of the courthouse separated from public access and the general public from the outside such as spectators or reporters. All access from the public area to the judge's chamber should be checked by a security guard or a hardware-checking system such as a key lock. The zone of restricted access is referred to as the "restricted zone," and the zone for general access from the outside is referred to as the “public zone” in the *U.S. Courts Design Guide*. More restricted spaces, those requiring more rigorous security control, are those for prisoners or prisoner transport. For security reasons, these areas should be in a closed circulation with access allowed only to spaces for trials such as the courtroom and holding cells. The zone for prisoner’s movement is referred to as the “secure zone.”

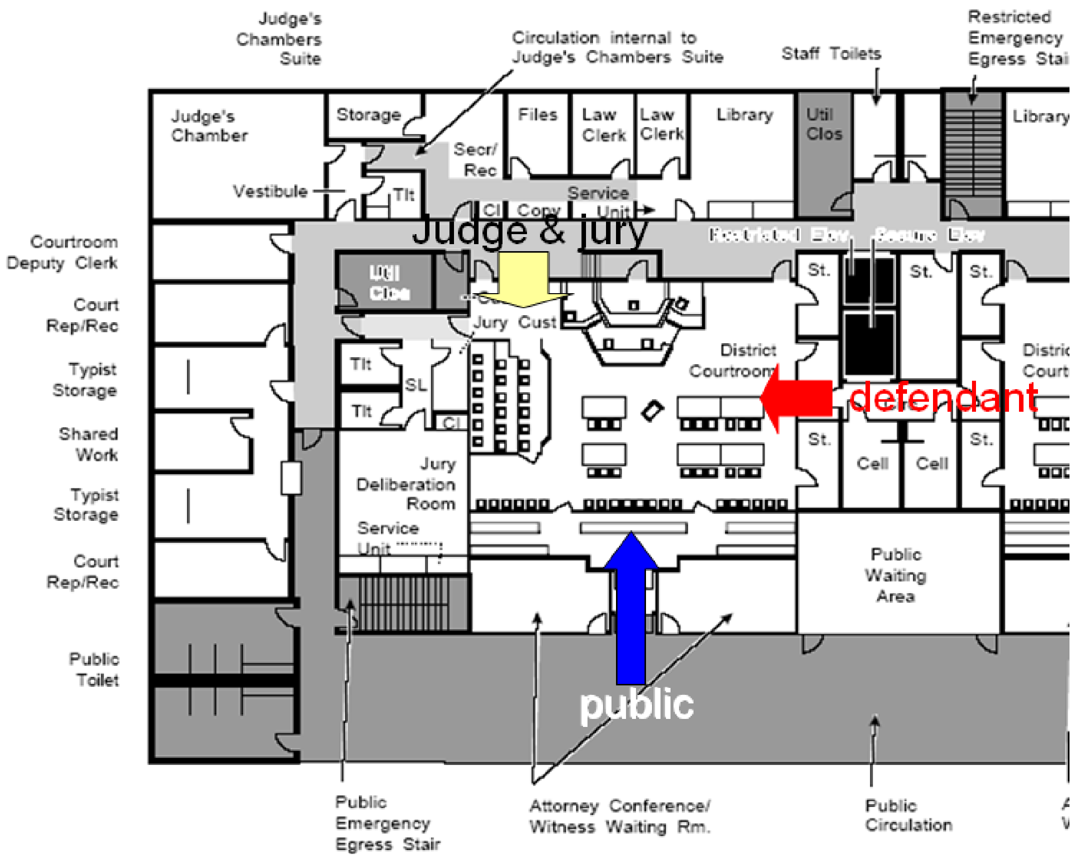


Figure 8 Example of a courthouse with three distinct security levels[2]

For security in movement through these three distinct security zones, *the U.S. Courts Design Guide* has a number of rules pertaining to connectivity among these spaces. The following section presents a review of these rules.

---

[2] From U.S. Courts Design Guide

3.2.2 Occupant circulation rules that govern the design of a courthouse

Reviewing a design guide or building codes is a critical step in the effort to determine what occupant circulation rules are and what the requirements of an occupant circulation checking system should be. Well organized occupant circulation for different types of people such as judges, prisoners, and spectators is critical to the security in a courthouse. Thus, the *U.S. Courts Design Guide* has many occupant circulation rules for the various individuals who visit a courthouse. From the review of the guide book, occupant circulation rules were classified to according to the requirements for a space model that supports the automatic checking of the rules. The rules can be classified into the following types: rules associated with circulation within a space and those related to circulation between spaces (see Figure 9).

- **Rules associated with circulation within a space**

Occupant movement takes place in large spaces such as the courtroom. Because occupant movement from one position to another position within that space can create problems in circulation, the *U.S. Courts Design Guide* lists several rules on the occupant circulation in large spaces such as a courtroom. The following is an example of a rule that governs circulation between a witness box and the spectator area in a courtroom.

Rule for Example 3

The witness box should be designed to allow the accessibility of a disabled person from the spectator area.

- **Rules associated with circulation between spaces**

Most rules associated with occupant circulation relate to connection between spaces. In contrast to the circulation rules within a space, those between spaces affect the global layout of the spaces in a building. Numerous rules in the *U.S. Courts Design Guide* relate to the interrelationship between spaces. The following is an example of a rule pertaining to circulation between the judge's chambers to the courtroom.

Rule for Example 4

The judge's chambers are accessed from restricted circulation to the courtroom.

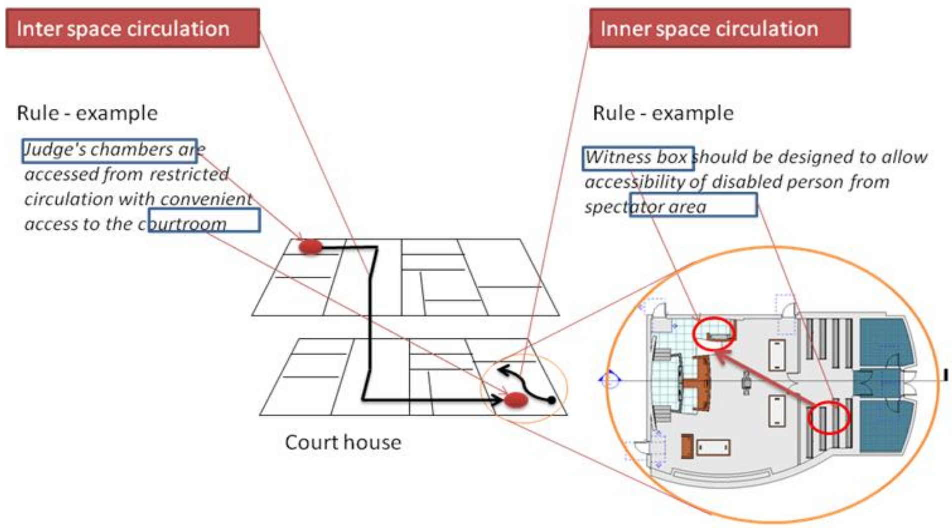


Figure 9 Inter-space circulation vs. inner-space circulation

The occupant circulation rules can also be classified according to the dependency of the shape of spaces.

- **Rules depending on the shape of spaces**

Some rules require details about the configuration and dimensions of spaces that need to be checked. One example is a rule that dictates the height of a space. Such a rule would require very specific height information. Example rule 5 is such a rule, for it requires specific dimensions of the area and ceiling height of the USDC jury trial space. This thesis refers to such rules as "geometry-related rules" or "geometric rules" in short.

Rule for Example 5

USDC jury trials require a courtroom of approximately 2,400 NSF (223 Nm2) and a ceiling height of 16 feet (4900 mm) (4-40)

- **Rules independent from the shape of spaces**

Some rules can be checked by using only the connectivity of spaces, which is independent from the shape of spaces. For instance, the example for rule 4 can be checked using the connectivity of the two spaces. This connectivity of spaces is independent from the shape of the paths among or between spaces. It simply focuses on whether spaces are directly connected or not. This thesis refers to such rules as "connectivity oriented rules" or "connectivity rules" in short**.**

These four types of occupant circulation rules have a different impact on building design, and the time that the rules are checked could vary according to the type of rule. Inner space circulation rules require more details of interior spaces such as furniture and fixtures. However, detailed interior information typically becomes available only at the end of the design stage. In most cases, interior design is independent from the space layout on the whole building level. Thus, the impact of inner space circulation is limited to individual spaces instead of whole building design. In contrast to inner space circulation rules, inter-space circulation rules are closely related to space layout on the whole building level, a global issue in building design; thus, it should be determined in the early stages of the design process.

Of course, inner space circulation may be more critical in certain building types such as theaters and concert halls. In these buildings, circulation inside the inner space may be more important than it is in other types of buildings. However, buildings that consist of several departments such as hospitals and airports require well-organized space layouts that support diverse circulation patterns among the departments. This thesis focuses on buildings that consist of multiple departments and zones in which occupant circulation is more important than it is inside of a single space.

Geometry-related circulation rules also could vary according to objects that may fall under the requirements spelled out in a particular rule. That is, a rule could be related to the shape of a door handle for accessing a room or associated with the shape of a space for measuring a route-distance. Some objects such as door handles and hand rails may become available in the later stages of design in general because they are detailing objects. However, rooms and corridors are decided in the relatively early stages of design because their shapes are required for estimating area and orientation.

3.2.3 The scope of occupant circulation rules for this thesis and their definitions

The scope of this thesis, which focuses on mainly inter-space circulation and connectivity-oriented rules, was determined based on their greater impact on the whole building level space layout as opposed to their relatively lesser impact on individual space. Partly geometry-oriented rules are included when they are required for connectivity checking. Inter-space circulation rules regulate the connectivity of spaces, which is becoming more complex and bringing up more complex issues associated with design development. Issues with inter-space circulation could lead to major design changes, or in some cases, they could scrap an entire project because they might be too costly or complicated to fix. Thus, inter-space circulation must be checked during the course of design development beginning at the early design stage.

As classified above, inter-space and connectivity-oriented circulation rules describe the requirements of circulation with connectivity between spaces. Inter-space and connectivity- oriented circulation rules mainly focus on occupant movement through connected spaces. (Other conditions than movement such as visibility and sound-proofed movement are beyond the scope of this thesis.) This thesis classifies these spaces into two types: terminal spaces and transition spaces. Terminal spaces are spaces located at the end of connections, and transition spaces are spaces placed in the middle of the connections. Figure 10 is a conceptualization of inter-space and connectivity-oriented rules pertaining to terminal spaces (i.e., a courtroom and a judge's chamber) and a transition condition (i.e., a restricted circulation) in transition spaces. The requirements for terminal spaces and transition conditions could vary in the set of diverse rules, and an investigation of the diverse requirements is one of main research objectives of this thesis.

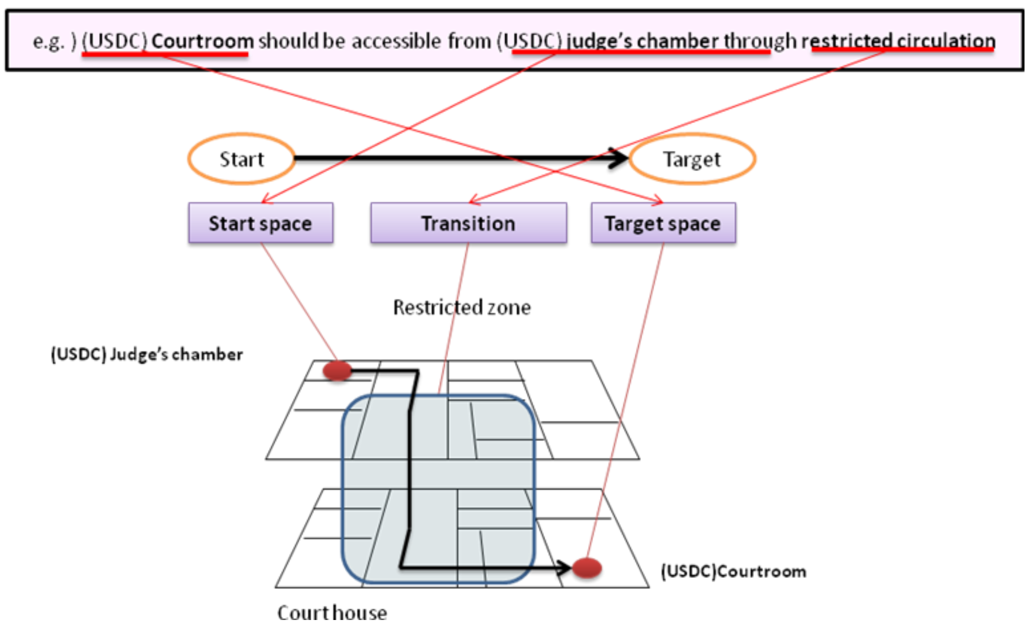


Figure 10 Conceptualization of inter-space and connectivity-oriented rules

This thesis describes the required conditions of the occupant circulation rules, the representation of the conditions as parameters, and the possible interpretation of a rule using the parameters.

3.3 Parametric representation of the circulation rule

The first step for the automatic checking of occupant circulation rules is the parameterization of circulation rules in the *U.S. Courts Design Guide*. In the checking of the circulation rules written by natural language, the rules could be interpreted using eleven parameters: *route cardinality, start space cardinality, target space cardinality, start space, target space, target space usage, zone condition, required space condition, direct access condition, vertical access condition, and length condition*. Identifying these parameters from the rules in the *U.S. Courts Design Guide* required the following procedure: 1. selection of the circulation rules, 2. classification of the circulation rules, 3. conversion of the rules into a normal form to find parameters, and 4. representation of the rules with parameters. This process is described in more detail in a reference "*Technical report-parametric representation of circulation rules*" [40]. As described above, inter-space circulation rules consist of a start space, a target space, and a transition condition between them. Thus, the ideal process of checking with a building model is both finding out whether the start target space and route between them in the model and checking whether the route satisfy the transition conditions or not. For instance, the rule in Figure 10 can be checked by finding the judge's chamber, the courtroom, and the route between them and then checking the satisfaction of the transition condition of the route. However, in general, a courthouse has multiple judge's chambers, courtrooms, and routes between them. Thus, the following questions may arise: How many target spaces should be checked from a start space? How many routes should satisfy the route conditions between them? Even if these conditions are not explicitly described in the circulation, these conditions should be defined clearly for automatic rule checking. Thus, this is a problem caused by applying a space in a rule to multiple instances of the space in a model. This thesis refers to such issues as "circulation cardinality issues," or "cardinality issues" in short.

3.3.1 Cardinality issues for circulation rule checking

Cardinality issues in circulation rule checking pertain to the number of paths needed in order to satisfy given conditions (i.e., valid paths) between a start and a target space. An "*all cardinality*" condition indicates that the results of checking a rule are true only when all paths between the start space and the target space satisfy the given conditions. The "*At least one*" cardinality condition indicates that the results of checking a rule are true when at least one path satisfies the given conditions. However, these cardinality issues become more complicated when multiple instances of a start and a target space are found in real buildings.

In order to describe cardinality issues more logically, the start space and target space are idealized in the Figure 11.

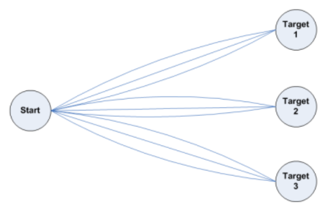


Figure 11 Idealization of routes between a start and target spaces

One start space may have multiple target spaces, and a start space and a target space may have multiple routes between them. The number of target spaces that satisfy given conditions is defined as "target space cardinality," and the number of routes between two spaces that satisfy given conditions is defined as "route cardinality." For example, Figure 12 shows that in the simplest case that satisfies the cardinality conditions, both target space cardinality and route cardinality are "at least one."

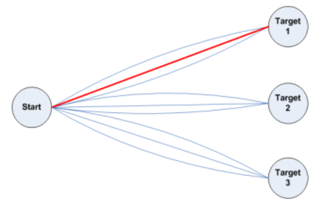


Figure 12 Target space cardinality: at least one; route cardinality: at least one

The following rule (6-19-16) is an example of a case that belongs to this category of cardinality.

"*Judges must have access to toilets convenient to the courtroom and accessible through restricted circulation (6-19-16)*."

If this route has at least one toilet accessible from the courtroom, and at least one route connects the toilet and the courtroom, which are both accessible through restricted circulation, then the circulation between these two spaces satisfies the rule.

Figure 13 shows a simple case in which target space cardinality is "all" and route cardinality is "at least one."

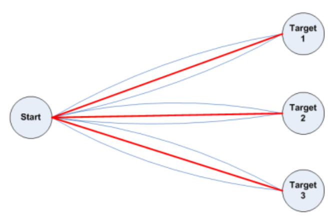


Figure 13 Target space cardinality: all; route cardinality: at least one

For example, the rule "*The courtroom should be accessible from the audio, video equipment storage through restricted circulation (3-16-27)"*represents such a case. This rule is true when at least one route in restricted circulation connects an audio/video equipment storage and each courtroom in a courthouse. Thus, this rule is an example of the cardinality condition in which the target space cardinality is all and the route cardinality is "at least one."

Figure 14 shows a simple case in which target space cardinality is “at least one” and route cardinality is “all.”

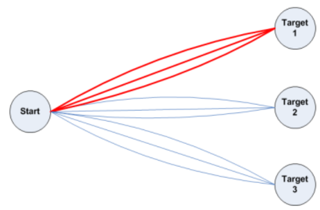


Figure 14 Target space cardinality: at least one; route cardinality: all

For example, the rule “*The USDC courtroom should be accessible from the Prisoner HLDG.CELL only through secure circulation (3-14-16)*”represents such a case. This rule is true when the USDC courtroom has at least one accessible prisoner HLDG Cell (target space cardinality), and when all routes between the USDC courtroom and the prisoner HLDG Cell are in the secure zone (route cardinality). Thus, this rule is an example of the cardinality condition in which target space cardinality is “at least one” and route cardinality is “all.”

In this case, the start space should be the USDC courtroom and the target space the HLDG Cell because the USDC courtroom should have at least one accessible HLDG Cell. If the order is changed, then it would not represent a case in which the USDC courtroom does not have an accessible HLDG cell.

Figure 15 shows the most rigorous cardinality condition. In this case, it is true only when target space cardinality and route cardinality are both “all.”

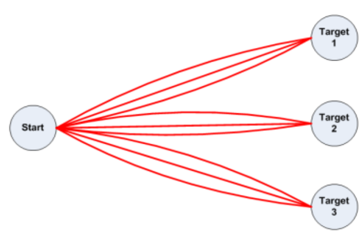


Figure 15 Target space cardinality: all; route cardinality: all

For example, the rule “*The prisoner HLDG Cell should be accessible from the central HLDG Cell through secure circulation (14-8-11)”* represents such a case. This rule is true only when all prisoner HLDG cells are accessible from the central HLDG cell (target space cardinality), and when all routes between the central HLDG cell and the prisoner HLDG cell are in a secure zone (route cardinality). Thus, this rule is the example of the cardinality condition in which both target space cardinality and route cardinality are “all.” The following section discusses the meanings and the uses of the parameters for defining circulation rules, including the concept of cardinality with examples of parameterized circulation rules. Since these parameters are likely to pertain primarily to courthouses, a broader set of rules will be required for general circulation testing in other building types. However, thesis focuses on rules in U.S. Courts Design Guide, and following are the conditions found in Occupant circulation rules for Court.

3.3.2 Parameters for representing circulation rules

Route cardinality:

Route cardinality defines a condition on the number of routes between a start space and a target space. “All” cardinality indicates that this rule is true when all routes between a start space and a target space satisfy a given condition. “At least one” cardinality indicates that the rule is true when at least one route between a start space and a target space satisfy a given condition. (A more detailed explanation of the cardinality issues follows in “Cardinality issues for circulation rule checking”**).** The default value for route cardinality is at least one because it is the most common in the circulation rules.

<e.g.>: cardinality (“zero”|”one”|”at least one”|”all”)

Target space cardinality:

Target space cardinality defines a condition on the number of target spaces that satisfy a given condition. “All” cardinality indicates that this rule is true when all target spaces have a route or routes that satisfy a given condition. “At least one” cardinality means that the rule is true when at least one target space has a route or routes that satisfy a given condition. (A more detailed explanation of cardinality issues follow in “Cardinality issues for circulation rule checking”).The default value for target space cardinality is at least one because it is the most common in the circulation rules.

<e.g.>: cardinality (“zero”|”one”|”at least one”|”all”)

Start space:

A start space should be distinct in the class level, but it does not need to be distinct in the instance level. For example, in the case of four types of courtrooms—a USDC courtroom, a USBC courtroom, a USCA courtroom, and a magistrate judge courtroom—these four types with distinct names should all be labeled “courtroom.” They do not need to be distinguished by instance level such as “USDC courtroom-01, USDC courtroom-02…”

<e.g.>: Start space (“USDC courtroom”|”USBC courtroom”|”USCA courtroom”|……)

Target space:

A target space should also be distinct in the class level, but it does not need to be distinct in the instance level.

<e.g.>: Target space (“USDC courtroom”|”USBC courtroom”|”USCA courtroom”|……)

Zone condition:

A zone condition defines the security level of the transit space. A default value for a zone condition is “Null” when it is not specified in a rule.

<e.g.>: zone (“public”|”restricted”|”secure”)

Door condition:

A door condition is for checking “controlled access” in circulation rules. “Controlled access” is interpreted as any access through a door with lock hardware. However, the model in the concept design level does not contain door hardware information. Thus, currently all access through different zones is regarded as controlled access, and all doors between the various zones are assumed to have a door lock. The default value for a door condition is “Null” when it is not specified in a rule.

<e.g.>: doorCondition (" "locked" | "controlled" ")

Required space condition:

A required space condition defines a required space that should be on a route. If a required space is defined in a required space condition, then at least one required space should be on a route. The default value for a required space condition is “Null” when it is not specified in a rule.

<e.g.>: requiredspacename (“screening point”|”…..”|”…..”)

Direct access condition:

A direct access condition defines direct accessibility between two spaces. The direct access condition is true when the number of transition spaces is zero, except the case in which the type of space is an entrance such as a vestibule. The default value for a direct access condition is “Null” when it is not specified in a rule.

< e.g.>: directAccessCondition (" "direct" | "indirect" ")

Vertical access condition:

A vertical access condition defines accessibility through vertical access elements such as stairs and elevators. Some rules include an accessibility condition on the same floor. In this case, this condition is checked as “not allowed.”   The default value for a vertical access condition is “Null” when it is not specified in a rule.

<e.g.>: verticalAccessCondition (" "allowed" | "not Allowed" ")

Length condition:

A length condition checks the distance between two spaces. The default length for a zone condition is “Null” when it is not specified in a rule. It defines the maximum length allowed.

<e.g.>: routeLength (" <comparator> "," <number> ")

Space usage condition:

A space usage condition defines whether transit spaces can generally be used for circulation or not. For example, a corridor is generally used for circulation, but private offices are not. This condition is not explicitly defined in the rules, but most rules imply this condition. In most cases, this condition is defined as “circulation” because a route through a circulation space such as a corridor or a lobby is reasonable. However, sometimes we need to set this condition as “private,” which applies to all non-circulation spaces such as restrooms or offices.   For example, when we check whether any other route does not pass the screening checkpoint from the main entrance to the inside of the building, the pass through the private space also should be checked for security. In that case, this condition should be defined as “private.”The default value for a space usage condition is “circulation” because it is an implication of most circulation rules.

< e.g.>: usage (“circulation”|”private”)

3.3.3 Example of an interpretation with parameters

By using the conditions described above, the Occupant Circulation Rules in the *U.S. Courts Design Guide* are interpreted into a computerized form. In the interpretation, a start and a target space should be selected with consideration of the application of the rule to real building design even if the rule implies bi-directional accessibility. In general, a building has many instances of spaces for a start and a target space defined in a rule. In order to apply a rule to a real building, a corresponding target space or spaces to a start space must be known. In general, spaces in a building model do not have information on the pairing of a start and a target space for circulation checking. Thus, parameterization of a rule with a start space and a target space should be selected considering the pairing of spaces in checking.

This thesis assumes that a route traversing begins from a start space and goes to a target space until the traversing finds a route that satisfies the required conditions in a given rule. In this approach, the start space has “all cardinality” because all start spaces in a model are checked versus the number of target spaces, and the number of valid routes to be checked depends on their cardinality conditions.

This process repeats until it checks all the instances of a start space in a rule. In other words, this process checks all the instances of a start space in a rule, but not all the instances of a target space if a rule requires just one target space for any start space.

In the interpretation of a rule, the space which requires the checking of all instances should be a start space and the other should be a target space. For instance, the rule “A judge’s chamber should be accessible to the courtroom through a restricted zone” should be interpreted to have a judge’s chamber as a start space and a courtroom as a target space. If a courtroom is defined as a start space, then it may cause a problem because it could result in a judge’s chambers not being checked.

However, it is possible to assign a cardinality condition to a start space theoretically. If there were a rule “There should be at least one office that is accessible to storage directly in a building.” It would contain at least one cardinality for a start space—an office accessible to storage satisfying the direct access condition. Thus, the start space cardinality condition is added to the circulation conditions.

The rules on occupant circulation are interpreted using the following eleven conditions: a start space, start space cardinality, a target space, target space cardinality, route cardinality, a space zone, space usage, direct access, vertical access, and distance. From these eleven conditions, we formulated 302 circulation rules, examples of which follow:

Example rule 1:

> *The judge's chambers are accessed from restricted circulation with convenient access to the courtrooms.*

Interpretation:

Start space: In general, a judge has a court, so the main space that must be checked is the judge's chamber, which should have access to at least one court. Thus, the judge's chamber becomes a start space. Courthouses usually house four types of judges: a U.S. District Court judge, a Senior U.S. District Court judge, a bankruptcy judge, and a magistrate judge. To maintain a distinction among the types of judges and types of courtrooms, a specified name of the judge and courtroom with the type are used.

Start space cardinality: Each judge's chamber in a building should have at least one courtroom that is accessible through a restricted zone. All start spaces should be tested without exception. Thus, this rule has "all" start space cardinality for all rules.

Target space: A target space should be a courtroom corresponding to the type of judge's chamber. For instance, a target space for a U.S. District Court Judge's chamber should be a U.S. District Courtroom.

Target space cardinality: This rule implies that one courtroom is enough for one judge's chamber. Thus, it is parameterized as "at least one."

Route cardinality: This rule implies that there should more than one route that allows access from the judge's chamber to the courtroom, but it does not require that all routes be in a restricted zone. Thus, route cardinality is parameterized as "at least one."

Space zone: This rule requires accessibility through a restricted zone. Thus, it is parameterized as "restricted."

Space usage: This rule requires access through circulation spaces such as a corridor instead of a non-circulation area such as a toilet. Thus, it is parameterized as "circulation."

Required space: This rule does not require a specified space that should be on a route between the judge's chamber and the courtroom. Thus, it is parameterized as "n/a."

Direct access: This rule does not require direct access. Thus, it is parameterized as "n/a."

Vertical access: This rule specifies that a route between the judge's chamber and the courtroom could go through an elevator or stairway. Thus, it is parameterized as "allowed."

Distance: This rule requires "convenient access." The distance can be used as a factor to measure how convenient access is. However, its quantification is beyond the scope of this thesis. Thus, it has a column to define a distance, but the value of the maximum distance is at the discretion of the checker. The distance is recorded so that it can be checked manually by a checker.

Parametric Interpretation[3]

Cardi-nality	Start space	Target Space cardinality	Target space	Security Zone	Usage	Direc-tion	Required space	Direct access	Vertical access	Dis-tance
At least one	USDC Judge chamber	At least one	USDC courtroom	Restric-ted	Circu-lation	n/a	n/a	n/a	allowed	Given value

The following is another example of the parameterization of a rule with a required space condition.

Example rule 2

The public must enter the court building lobby through a single security screening point, controlled by court security officers.

Interpretation:

This rule implies that all routes between the main entrance and a public lobby should have a screening point. Thus, both route and target space cardinality should be “all.” The start space is the “main entrance” and the target space “lobby.” The public area is between the main entrance and the screening checking point. Thus, the security zone parameter is set as “public.” The route between them includes a route through the non-circulation space also because the purpose of this rule to find a route that does not pass through a screening point. The following table shows the parameterization of the rule based on the interpretation.

Parametric Interpretation

Route Cardi-nality	Start space	Target Space cardinality	Target space	Security Zone	Usage	Direc-tion	Requir-ed space	Direct access	Vertical access	Dis-tance
All	Main entrance	All	lobby	public	Circu-lation	n/a	Scree-ning point	n/a	allowed	n/a

[3] Start space cardinality condition is omitted because it is always “all” for circulation rules in U.S. Courts Design Guide.

CHAPTER 4

GRAPH AND SET BASED CHECKING

4.1 Overview

This section describes the currently used methods developed for checking the occupant circulation rules in the *U.S. Courts Design Guide*. It includes two methods, a graph-based method and a set-based method, which are developed for the checking of a building model during specific design stages (the late concept stage and the early concept stage).

4.2 Two levels of the design stage (final concept design and preliminary concept design)

Public Building Service Design Guide - PBS Design Guide [44] defines its own design process and specifies the time for assessing a design according to its guidelines. It specifies many stages in which the design should be checked and defines the Preliminary concept design and the Final concept design, one type of building model in the design process for checking. A design in the early checking stage is referred to as a preliminary concept design and a design in the late checking stage a final concept design. A typical model in the preliminary concept design is made of aggregated spaces that have not yet been developed into individual spaces, and the model has no doors that explicitly connect the spaces, only zones defined on the departmental level. The final concept design consists of individual spaces with doors that explicitly represent space connection. This research has developed two methods that support these two different models: a graph-based method for the Final concept design and a set-based method for the Preliminary concept design. The following sections explain the two methods.

4.3 Requirements for an automatic checking module

Now that the rules on inter space circulation in the *U.S. Courts Design Guide* have been reviewed, the requirements for rule checking system will be analyzed as follows.

- **Space connectivity information for circulation rule checking.**

Not all data in the IFC are required for checking occupant circulation rules on space connectivity. The checking of the more than 300 parameterized rules will be more efficient if a data structure that contains diverse space connectivity rules is used rather than the entire IFC dataset. Many previous studies have used graphs to represent space connectivity. This thesis also adopts a graph for checking circulation rules.

To interpret the circulation rules in the *U.S. Courts Design Guide,* the graph should be able to represent the eleven parametric conditions defined in the previous chapter. It should also be able to traverse between terminal spaces to find a correct route and check diverse parametric traverse conditions such as the zone condition, the required space condition, the direct access condition, and the vertical access condition. In addition, it should be able to check the number of correct routes between terminal spaces to assess cardinality conditions. The graph can also represent other conditions described below.

- **Space grouping**

Among various rules, some rules could be checked using containment in a specific group. For example, a rule pertaining to a toilet for the clerk's office, -i.e., *Toilet should be placed in the same floor where clerk office is.,* can be used to check whether a toilet and a clerk's office are located on the same floor. Containment on the same floor could also be checked using the routes between them. If they are connected by a route that has no spaces for vertical circulation such as elevators, a stairway, or a ramp, then it means these two spaces are on the same floor. This approach uses an inverse condition of a required space on a route. If a route contains no spaces for vertical circulation, then it means that the two spaces are connected by a route on the same floor. Another solution for this rule is to use a grouping of spaces by floors. With space grouping by floor, the checking of group membership of two spaces on a floor group can be used for containment checking. Checking using grouping information is more efficient because it does not require traversing many routes between the terminal spaces. The checking by routes could be an expensive process if a building consists of many spaces. A building could have more than one hundred stories with the number of rules that have to be checked also exceeding a hundred. In such a case, Checking time could be saved by both filtering out some rules that could be checked by grouping and checking the rules by containment checking. The concept of grouping is not limited to only floor grouping. In general, spaces in a building consist of a number of grouping concepts such as zones, departments, and aggregations. These groupings are subject to several occupant circulation rules such as *All routes between the central holding cell and the local holding cell should be in a secure zone only.* Thus, grouping as a pre-checking method before performing route-based checking may improve the performance of checking, particularly that in large-scale buildings.

4.4 Graph-based circulation rule checking

4.4.1 Representation of a spatial configuration with a graph

In general, a space in a building is a discrete space bounded both physically and functionally. A space is connected to other spaces through connectors such as doors, openings, stairs, or elevators. This feature distinguishes a building space from a street or city space, for which boundaries are difficult to define. Spaces in a building have relatively clearer boundaries than spaces on the outside, and it is accessible to the other spaces through connections. A graph for circulation rule checking is comprised of nodes and edges for spaces and connectors, explained below.

- **Nodes for spaces**

A space in a building indicates a volume enclosed by a physical boundary such as a wall, a slab, or a functional boundary such as a virtual wall. As figure 16 shows, a space could consist of all physical boundaries, or it could be bounded by all functional boundaries. A node for a space is placed within the boundaries of the space. A node for a space is referred to as a "space node."

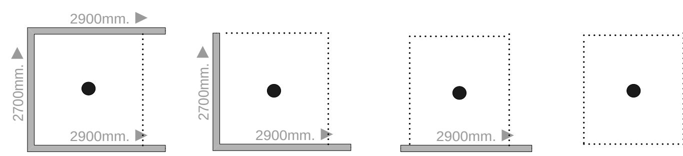


Figure 16 Diverse boundaries of a space and a space node for a space

- **Nodes for connectors:**

A connector is an open area in a space that allows a person to move to other spaces. It could be in a wall for horizontal access or in the floor for vertical access. An opening can be just an open area without any physical element, or it can be a physical element such as a door. Each space should have at least one opening that connects to another space for access. If a space does not have at least one opening, the space cannot be used for circulation. In case a space has only one opening, it cannot be used for circulation through it. If a space is divided into two functional spaces with a virtual wall, then the virtual wall is regarded as an opening that connects these two functional spaces.

A node for a connector is referred to as "connector node." A connector node on an opening, a door, or a virtual wall is an opening node, door node and virtual wall node, respectively.

Figure17 shows examples of a graph consisting of space nodes, connection nodes, and edges. A space can be connected by at least one connection node. The connection node could be an opening node, a door node, and a virtual wall node. Spaces separated by a physical or functional boundary should be connected through at least one connection node on the boundary. The graph made by a space and connector nodes is called a “space graph” (see Figure 18).

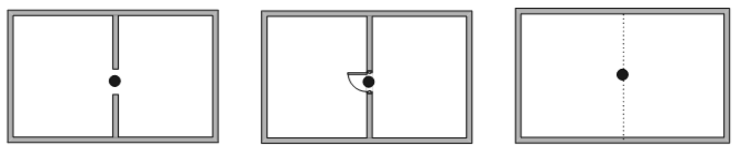


Figure 17Connector nodes: an opening node, a door node, and a virtual wall node

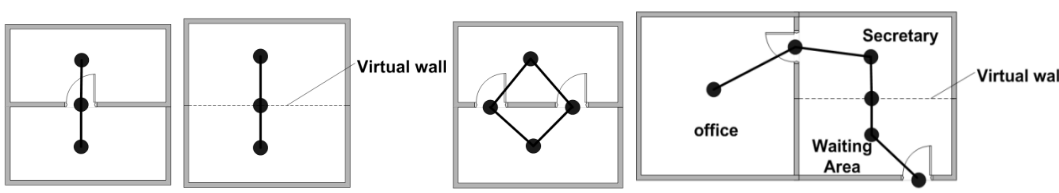


Figure 18 Examples of space graphs

4.4.2 Modeling a building with an IFC

An Industry Foundation Class (IFC), a public domain data model that represents building information, facilitates the interoperability of the building information during the whole building life cycle. The schema of the IFC was defined by buildingSMART and major BIM authoring software such as Autodesk Revit, ArchiCAD, and Bentley MicroStation, support IFC schema version 2X3.

Theoretically, a rule checking system should be independent from BIM tools and the modeling process. It should be able to cover the diversity in a model and to check all the rules. However, the scope of diversity in modeling is open-ended, and in practice, it is virtually impossible to cover all diversity in modeling. In other words, even though the checking system should allow a certain level of flexibility of modeling, in practice, it is necessary to define the modeling requirements for checking certain types of rules. For any set of rules, a corresponding set of modeling requirements must accompany the data needed for checking occupant circulation rules.

Table 1 Relationship between building elements and IfcEntities

Modeling	IFC Entity Name	Description of Modeling
Space	IfcSpace	▪ It is used for representing diverse spaces in a building such as offices, judge's chambers, courtrooms, staircases, and elevators. ▪ Spaces have a geometric shape, height, and volume. ▪ Shapes can be bounded not only by IfcWalls but also by a virtual boundary. ▪ Space names are used as identifiers of each space.
Door	IfcDoor	▪ It is used to represent a filler in a wall opening. ▪ Doors are embedded in a wall and identify a pass-through circulation condition. ▪ A door should be adjacent to at least one space and usually two.
Stairs	IfcStair or IfcSpace or both	▪ Every type of stair defined the IFC product model list can be supported by BIM vendors. ▪ Stairs can be modeled by IfcStair with the detailed shape of a stair, or it can be modeled by using IfcSpace with the spacename Stair or both may be used.
Ramp	IfcRamp or IfcSpace	▪ Every type of ramp defined on the IFC product model list can be supported by BIM vendors. ▪ It should be modeled by IfcRamp with the detailed shape of the IfcRamp, or it could be modeled by IfcSpace for the boundary space of a ramp or both.
Elevator	IfcSpace	▪ Elevator objects are defined with an elaborated space name (e.g., "judge's elevator" or "prisoner elevator"). ▪ IFC does not have IfcEntity for elevator car; thus, it should be modeled using IfcSpace for Elevator shaft.
Wall	IfcWall, IfcCurtainWall, IfcStandardCaseWall	▪ It is used to represent all internal or external walls. ▪ It should have a relationship with the openings in it.
Virtual boundary	IfcRelSpaceBoundary	▪ It connects a space with the physical or virtual elements that bound it. ▪ Doors typically have two IfcRelSpace boundaries. ▪ It is used to represent relationships between both the IfcWalland/or the IfcVirtualElement for a virtual wall. ▪ Must be defined for all doors, so they define access between spaces.
Opening	IfcOpeningElement	▪ It is used to represent any opening in a physical boundary, especially one with doors but sometimes without.

The building elements relevant to occupant circulation are spaces, doors, stairs, ramps, elevators, virtual boundaries, openings, and walls. The required information will be represented by IFC entities, and the rule checking system will use the IFC entities for checking. Table 1 has detail information on the modeling requirements.

4.4.3 Generating a graph from an IFC building model

Graph generation is a process of interpretation of the relation among spaces in an IFC building model with nodes and edges in a graph. This process is done by mapping IFC entities to nodes and by mapping the relationships between the IFC entities and the edges. IFC entities relevant to occupant circulation such as IfcSpace, IfcDoor, IfcStair, IfcRamp, IfcVirtualElement, and IfcOpening are mapped to a node in a graph, and the relationships between them are mapped as edges.

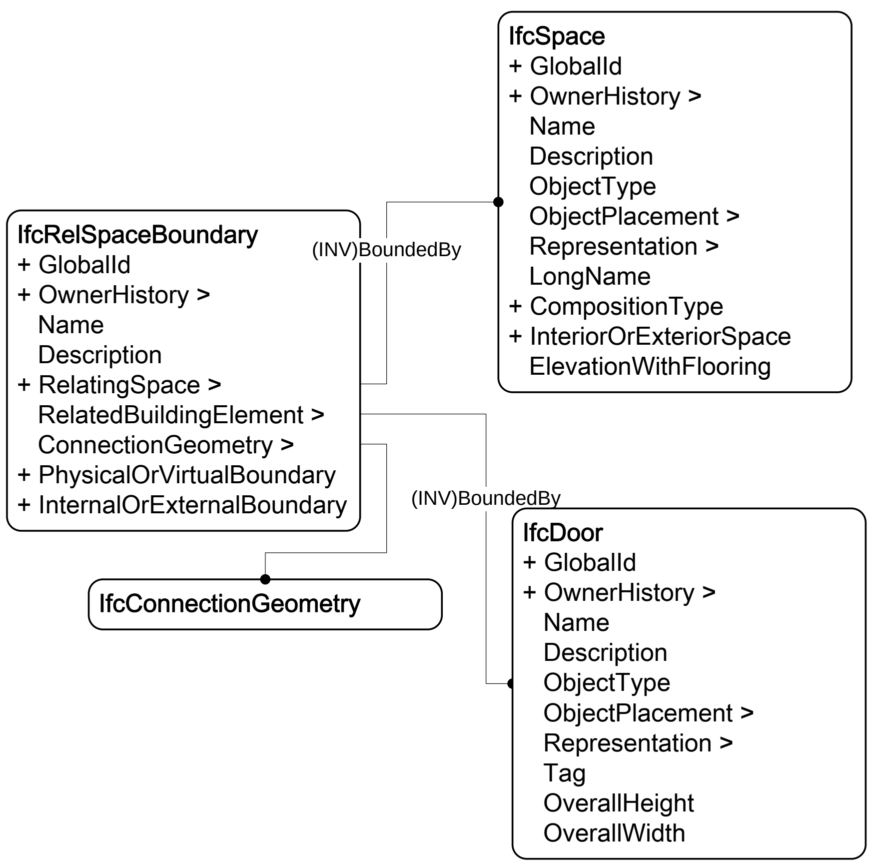


Figure 19 IfcConnectionSufaceGeometry between IfcSpace and IfcDoor

A relationship between a space and physical or virtual boundaries is defined by fcRelSpaceBoundary. IfcSpace could be connected to IfcDoor through IfcRelSpaceBoundary, and IfcConnectionGeometry, which represents the surface geometry of an intersection between a space and a door (see figure 19).

If two spaces are connected through a virtual wall, they are associated with IfcRelSpaceBoundary, and IfcConnectionGeometry represents the surface geometry of the virtual wall between them.

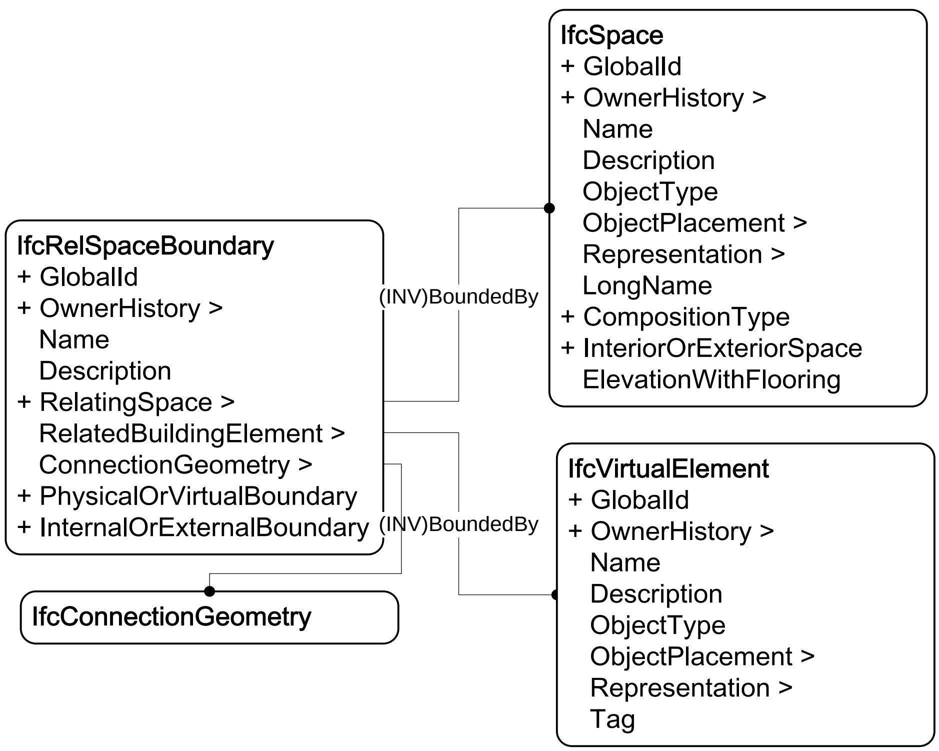


Figure 20 IfcConnectionSurfaceGeometry for a virtual wall

If two spaces are connected through a door, the door is associated with the two spaces through IfcRelSpaceBoundary. Thus, the connectivity of the two spaces can be retrieved by using the association (see figure 20).

IfcStair and IfcRamp are entities for vertical connection. They connect spaces vertically. Since there is no entity for an elevator in IFC, it could be represented by using IfcSpace for an elevator shaft. The relationships between vertical connectors and spaces are not defined explicitly in the IFC. Thus, the vertical connection through Ifcstair and Ifcramp, or Ifcspace for elevator should be calculated by using the location and the geometry of these objects. Table 2 shows the mapping between IFCEntities and graph elements.

Table 2 Mapping between IFC entities and graph elements

IFC Entities	Graph Elements
IfcSpace, IfcDoor, IfcStair , IfcRamp, IfcVirtualElement and IfcOpening	Nodes
IfcConnectionSurfaceGeometry IfcRelSpaceBoundary	Edges

Figure 21illustrates a space graph generated by mapping. This space graph consists of 24 spaces, which includes the judge's chamber, the toilet, the corridor, and so forth. The blue nodes in the graph denote spaces, and red nodes denote connections between spaces. The graph containing space nodes and connection nodes exhibits the following features.

A space node is connected to at least one door node except spaces without doors such as mechanical shaft.

All spaces accessible to any person should have at least one opening. Thus, all space nodes except for non-accessible spaces such as a mechanical shaft or an air duct should be connected to at least one door node.

All end nodes in this graph are space nodes except a door node for a main entrance.

A door node connects two space nodes. Thus, a door node could not be an end node except when a door node is a main entrance or an exit door to the outside.

All edges in the graph have one space node and one door node at each end.

Only an edge connecting a space node and a door node is valid. An edge connecting a space node to a space node is invalid because it indicates two spaces connected without an opening area. If a space has no opening, such as a mechanical shaft or a duct, it does not have any connection to other spaces because it does not have any openings to connect them. An edge for a door to a door is also an invalid connection because it indicates that two doors are connected without any space between them.

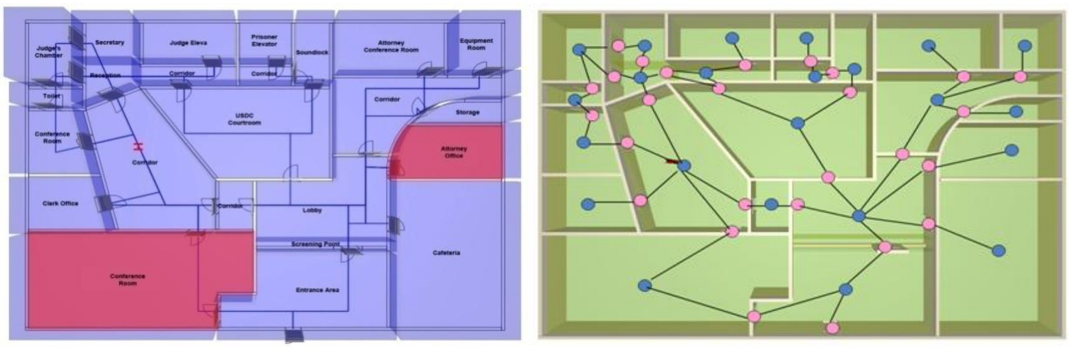


Figure 21 An example of a space graph

4.4.4 A logical process of checking circulation rules with a graph

The basic concept of occupant circulation checking is to find a route that satisfies all the required conditions. If two spaces are connected through a correct route that satisfies all the required conditions, the layout of the two spaces obeys the required conditions.

The logical process for checking a circulation rule consists of three steps: identifying all of the routes between the start and target spaces, filtering out routes through private spaces, and filtering out the routes that do not satisfy the required conditions. One must keep in mind that this is a conceptual process for easy understanding of the checking process based on routes. The actual process of checking considers the efficiency of checking, so the processes listed below are integrated and performed in the same time period, if possible.

Step 1: Identifying all of the routes between the start and target spaces

This process finds all the routes given the start and target spaces. All the routes between two spaces include a shortest path and a longest path. The first routes in Figure 22 are all routes between spaces A and B.

Step 2: Filtering out routes through private spaces

In general, movement from one space to another is done through circulation space such as corridors, stairs, or lobbies. Private spaces such as offices, conference rooms, or toilets, are not generally used for occupant circulation. Thus, the first filtering excludes all the routes through non-commutable spaces. The definition of commutable or non-commutable space depends on the type of building and the type of person who has access. For instance, a secretary's office of the manager is a commutable space for the boss. However, this space is not a "go-through" space for the other person. Thus, the decision on commutable spaces should be done with consideration of type of building and type of person. The second routes in Figure 22 are all the routes that remain after excluding the routes that pass through non-commutable spaces.

Step 3: Filtering out the route that does not satisfy the required conditions

This step determines whether the routes satisfy the required conditions or not. In the case of "all route" cardinality, if a route does not satisfy the given condition, then the layout of space A and B is wrong. In the case of "at least one" cardinality, if a valid route exists, then the layout is correct.

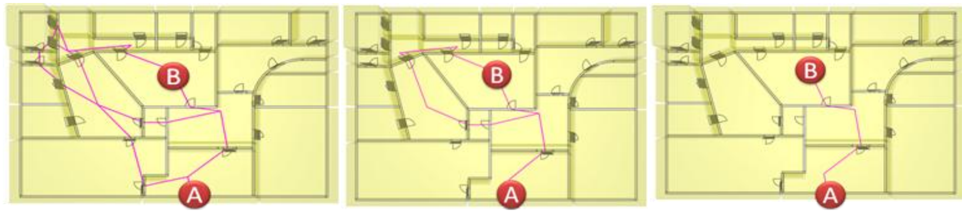


Figure 22 Logical process of route checking

In real implementation of the checking module, it does not find all the routes between two spaces while looking for a valid route. Finding all the routes is very costly because the number of routes increases exponentially as the number of nodes in a graph increases. Instead of finding all the routes, Dijkstra's algorithm [45] on a weighted graph is used to locate the valid routes. Dijkstra's algorithm traverses a graph by a depth-first search from the start space node until it finds the lowest weighted space node. If it finds a space node that does not satisfy a required condition, then it assigns a large weight value to the node. After that, it repeats the search until it finds the lowest weighted correct route between the start and target spaces.

Graph-based model checking is applied to real court house design projects: courthouse 1 and courthouse 2. The courthouse 1 is a six-storey building consisting of 1,003 spaces, including six courtrooms and eight judge's chambers, and the courthouse2 is a five-storey building with one basement floor. It has 432 individual spaces, and six courtrooms. Three hundred two parameterized rules are applied to these models, and many design errors are detected within 30 seconds (see figure 23).

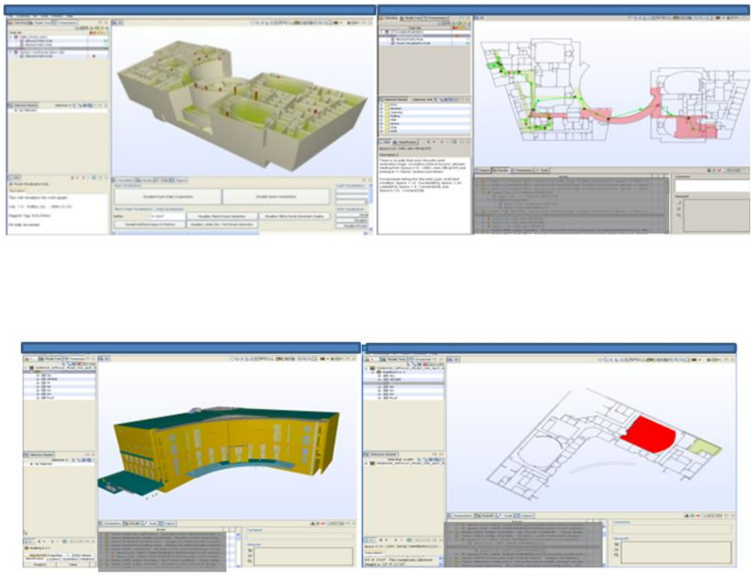


Figure 23 Checking of real courthouses by the graph-based method

Even though the graph-based checking method identified design issues from the real courthouse models in a short time, GSA noticed that it might have been too late to fix the issues because the model in the final concept stage was developed with full details. Changing the space connections at this stage would have been complicated because all the spaces were already developed with full connections.

GSA wishes to find design issues in the early stage of design, even if the spaces are not fully developed. Thus, a set-based circulation checking method is developed to check a model at an early stage.

4.5 Set-based circulation rule checking

4.5.1 Preliminary concept design

Early stage design checking is a crucial task because finding design errors in an already developed design is considerably more costly than fixing the errors in the early stage of design. The General Service Administration (GSA) has designated a stage during which checking can take place in the preliminary concept design, before a detailed design.

The second model in Figure 24 is a typical preliminary concept design of a courthouse. A preliminary concept design defines a space at a departmental or aggregated space level in detail. For instance, according to the *U.S. Courts Design Guide*, “probation” consists of many spaces such as the probation office, a reception area, a conference room, and a staff lounge. The early concept model uses department or aggregation space names such as “probation” instead of individual spaces.

Another feature is the lack of a door including a main entrance door or the lack of a door except the main entrance door only. As a model in Figure 24, the early concept model represents a space just by using boundaries of a space without physical elements such as walls. Thus, the early concept model does not have explicit information on the connectivity between spaces. Only the adjacency of spaces is available.

This thesis defines the design in the early stage as the Preliminary Concept Design and the design in the late stage the Final Concept Design.

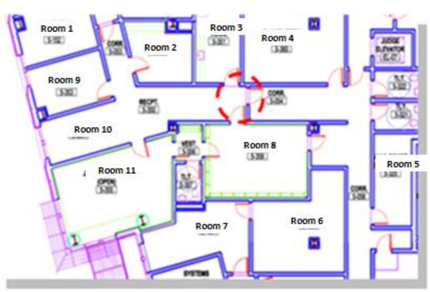
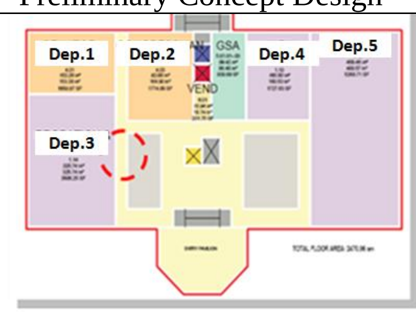
Final Concept Design	Preliminary Concept Design
	 [4]
Door, Individual space level definition	No door, Aggregation level space definition

Figure 24 Final concept design vs. Preliminary concept design

4.5.2 Set-based checking

Early stage design checking is a crucial task because finding design errors in an already developed design is considerably more costly than fixing the errors in the early stage of

[4] Space names in this figure are changed to general names such as room1, 2 and dep1, 2 for security reason.

design. The General Service Administration (GSA) has designated a stage during which checking can take place in the preliminary concept design, before a detailed design. The second model in Figure 24 is a typical preliminary concept design of a courthouse. A preliminary concept design defines a space at a departmental or aggregated space level in detail. For instance, according to the *U.S. Courts Design Guide*, “probation” consists of many spaces such as the probation office, a reception area, a conference room, and a staff lounge. The early concept model uses department or aggregation space names such as “probation” instead of individual spaces.

Another feature is the lack of a door including a main entrance door or the lack of a door except the main entrance door only. As a model in Figure 24, the early concept model represents a space just by using boundaries of a space without physical elements such as walls. Thus, the early concept model does not have explicit information on the connectivity between spaces. Only the adjacency of spaces is available.

This thesis defines the design in the early stage as the Preliminary Concept Design and the design in the late stage the Final Concept Design.

4.5.3 Set-based checking of circulation conditions

A door is used to generate a connection graph in order to make a connection between two spaces. Thus, the connection graph cannot be generated without a door. In other words, checking the circulation rule pertaining to the lack of a door means checking a circulation rule without a connection graph. In order to distinguish between the methods for circulation rule checking with a connection graph and that without a connection graph, the approaches are called a graph-based method and a set-based method, respectively. In the graph-based method, circulation rules with routes between terminal spaces are checked. In the set-based method, the containment of terminal spaces in a set consisting of a group of spaces with adjacency and in the same classification is checked. As shown in Figure 25, two terminal spaces are in the same zone would indicate a potential route between them in the same zone. However, if the two terminal spaces are located in separate zones, then no route connects them in same zone, indicating that some parametric conditions could be checked even if the early concept model does not have any information regarding space connectivity.

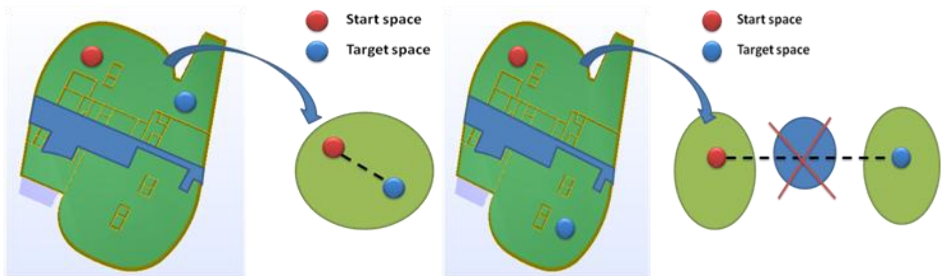


Figure 25 The set-based method

The following is the review of the parametric conditions, the purpose of which is to determine which conditions can be checked and which ones cannot be checked by the set-based approach.

The route cardinality condition

The route cardinality condition, which requires routes between the start and target spaces, cannot be checked by the set-based method.

The target space cardinality condition

The target space cardinality condition, which checks how many target spaces satisfy the required conditions, can be checked by the set-based method by determining how many target spaces are in a set.

The start space condition

The start space condition simply states the name of a start space, so it can also be checked by the set-based method. The type of space name could be either an aggregated space name or an individual space name. The checking of a space name by type is described in detail in Chapter 5.

The target space condition

The target space condition simply states the name of a target space, so can also be checked by the set-based method.

The zone condition

The zone condition of the transit space can be checked by the set-based approach if the rule has the “at least one route” cardinality condition. If two terminal spaces are within a same zone, at least one potential route could be in the zone. However, if two terminal spaces are not in a zone, then no route is within the same zone. If a rule has the “all target space” cardinality condition, then we need to check whether all target spaces are within the zone or not. If at least one target space lies outside the zone, then some routes do not satisfy the zone condition (Figure26).

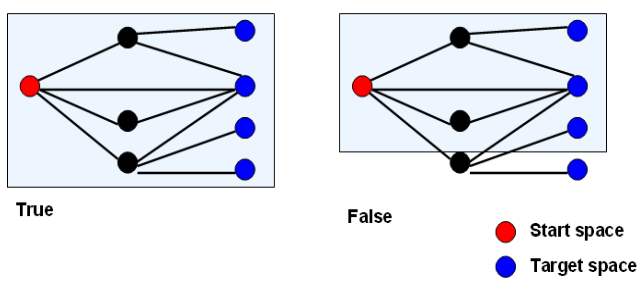


Figure 26 All target space cardinality checking

The circulation space condition

The circulation space condition indicates that any transit spaces in a route should be a circulation space, which depends on the route between two terminal spaces. For example, the route in case A [Figure 27] between a judge’s chamber and a courtroom satisfies the circulation space condition, but the route in case B [Figure 28] does not satisfy the circulation space condition because it goes through the jury assembly space for circulation. The set-based approach cannot check the circulation space condition because the set for the circulation conditions such as the sets for circulation spaces and private spaces does not explain the connection of routes between two terminal spaces.

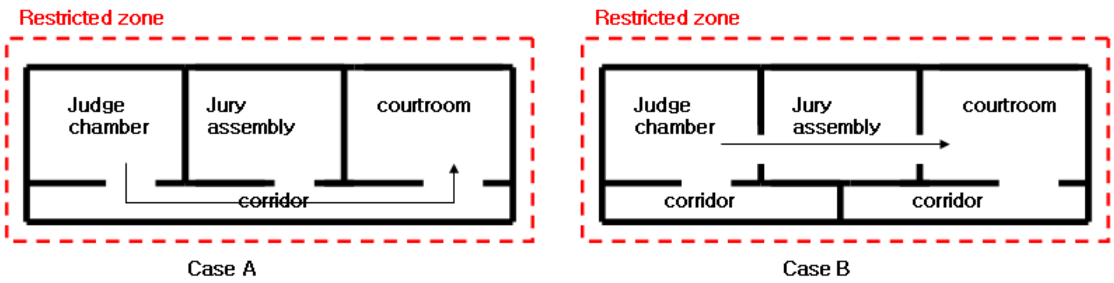


Figure 277 Circulation space condition

For instance, the spaces in Figure 27 can be classified into two sets: a set for private spaces (e.g., the judge's chamber, the jury assembly room, the courtroom), and a set for circulation spaces (e.g., the corridor). These two sets cannot explain whether a judge's chamber is connected to the courtroom through a circulation space or not.

The direct access condition

The direct access condition also requires route information because it is checked by counting the number of spaces between two terminal spaces. However, the direct access condition can be partially checked if each terminal space is placed in different groups and these different groups are clearly separated without any adjacency. Thus, the direct accessibility condition is partially checkable with the set-based approach.

The required space condition

If a start space, a target space, and a required space are in the same set, then a route with the required space is possible. If the required spaces are in a different set than the set with a start and target spaces, then it does not have the potential to have a route with a required space. Thus, this condition can be checked by set-based method.

The vertical circulation condition

The vertical circulation condition checks whether two terminal spaces can be accessible through a vertical circulation space or not such as an elevator or a stairway. If the spaces are classified by its floors, it can be checked by the set-based approach. If two terminal spaces are on the same floor, then they are accessible without passing through a vertical circulation space.

The distance condition

The distance condition requires a route to measure distance. Thus, it cannot be checked by the set-based method.

In summary, the zone and vertical space conditions with at least one route cardinality condition could be checked by the set-based method. Among the 302 rules, 142 (47%) can be checked by the set-based method.

4.5.4 Set-based representation of space relations

In order to check circulation rules with the set-based approach, spaces in a courthouse are grouped according to security level: public, restricted, or secure. Adjacent spaces on the same security level are merged until all adjacent spaces are merged and merged spaces become a set. The following diagram is a set-based representation of spaces in a courthouse. The edge between sets indicates adjacency instead of connectivity. There are two type of adjacency: horizontal and vertical. Horizontal adjacency is adjacency of sets on the same floor. Adjacency is only possible between spaces on different security levels because adjacency within the same security set is merged. In Figure 28, all edges connect different security zones: restricted–public and restricted-secure.

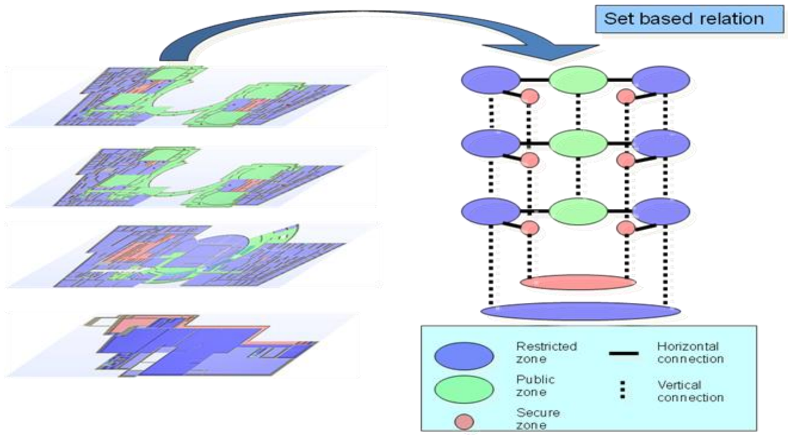


Figure 28 Set-based representation of space relations

However, vertical adjacency represents a vertical connection between two spaces through vertical connectors such as a stairway or an elevator. If two sets are connected through vertical connectors, then the two sets have vertical adjacency. Sets connected vertically can be merged and become a larger set. If a rule allows vertical access, a set merged vertically should be applied. The set-based method, which was applied to several real early concept models, found real design errors in the early design stage.

4.6 Summary

Although the graph-based method can check all the conditions pertaining to the occupant circulation rules found in the *U.S. Courts Design Guide*, it is applicable only if the graph can be generated from a model with information about space connectivity. Thus, it cannot check the model without connection elements such as doors, openings, or stairways. In practice, the design of a building is a continuous process between preliminary concept design and final concept design. Supporting the automatic checking of models in transition between preliminary concept design and final concept design is important for supporting the continuity of the design development. However, the graph-based method is limited to checking only for the final concept design -level model.

Even though the set-based checking method can check a model in the preliminary concept design stage, the checking algorithm depends completely on the sets, the aggregation of adjacent spaces with the same properties. It does not require any more information in a building model than the information the preliminary concept design model has. Thus, if the model is to be checked between the preliminary concept design and the final concept design, it should be simplified to a preliminary concept design model. However, such simplification could lead to a loss of existing information and inaccurate results of checking.

CHAPTER 5

GENERALIZATION OF CHECKING METHOD

5.1 Introduction

Building design in the developmental stage could vary from the diverse perspective of completeness. A design can have diversity in terms of completeness at the building entity level such as whether it has doors, walls, aggregated space, and so forth. It can also vary according to the completeness at the departmental level, floor level, and building system level such as the MEP system. From the almost infinite number of variations, the cases meaningful for occupant circulation rule checking should be investigated. The strategy of dealing with a lack of information in each case also should be studied.

To deal with the diversity of building models in the development stage in a more logical way, the requirements for occupant circulation rules are also described with logical symbols, and the checking process is described by the abstracted representation of building design and occupant circulation rules. Considerable research in computer science has adopted abstracted representation of algorithms of computation mainly derived from mathematics such as mathematical logic, computational geometry, graph theory, and so forth. This thesis selects predicate logic to represent the checking method to show the validation of checking logically. Predicate logic is adopted because it can not only represent the checking method logically but also deal with quantification issues in circulation rules, represent diverse conditions of circulation rules by using predicates, and handle building elements to be checked as an instance of predicate logic.

5.2 An abstracted representation of the circulation rule validation process

5.2.1 The general form of an occupant circulation rule

As reviewed above, occupant circulation rules can be interpreted by the start and target spaces and the transition conditions between them. Thus, this thesis defines the general form of occupant circulation rule as a combination of the three. For instance, in the rule “The courtroom should be accessible from the judge’s chamber through restricted circulation,” the “judge’s chamber” is the start space, the “courtroom” is the target space, and “accessible through restricted circulation” is the transition condition between them. Transition conditions are a combination of many conditions of traversing. In this case, it is a combination of two conditions; “through restricted circulation” explicitly and “through circulation spaces” implicitly. Thus, the process of checking, finding a start and a target space from a building model, is first. If they exist, then next step is finding a route satisfying the transition conditions. For instance, rule 1 can be checked by finding a judge’s chamber and a courtroom first. If they exist, then the next step is to find a route between them that satisfies the transition conditions. This thesis selects the start space, the target space, and the transition conditions as key elements for the general form of occupant circulation rule, and diverse types of occupant circulation rule are converted to fit to this general form (see Figure 29).

Rule 1: *The courtroom should be accessible from the judge’s chamber through restricted circulation.*

Start Space	**Target Space**	**Transition Conditions**
Judge’s chamber	**Courtroom**	**Access through restricted circulation**

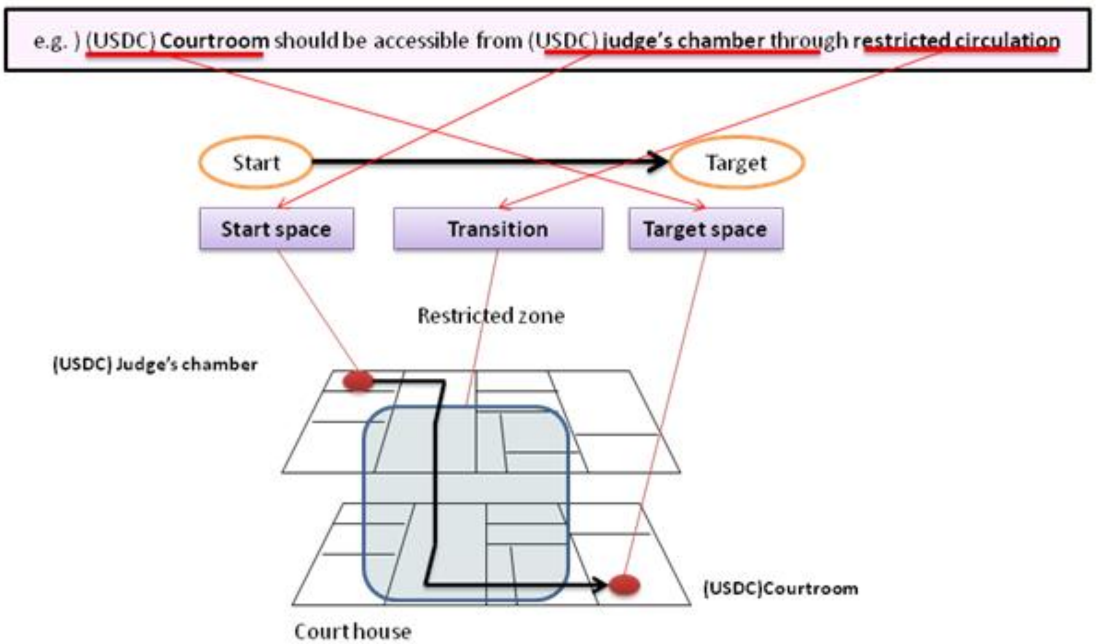


Figure 29 Process of circulation rule checking

Based on the general structuring of an occupant circulation rule, eleven parametric conditions are classified by three conditions: the start space condition, the target space condition and the transition condition.

Table 3 Parametric conditions

Start Space Condition		Target Space Condition		Transition Condition						
(Start space cardinality condition)	Start space	Target Space cardinality	Target space	Route Cardinality	Security Zone	Space Usage	Required space	Direct access	Vertical access	Distance

5.2.2 An abstracted representation of rule validation

An abstracted representation of circulation rule checking is developed to be a logical basis of a circulation checking method. An abstracted representation is based on logic notation, especially predicate logic, to deal with quantification issues in circulation rules and instance-level validation, which means the validation of individual building elements in building design. This approach is based on the following assumptions.

Assumption 1: *The validation of a circulation rule is the process of apply a rule to a building element or elements*

In general, rule checking is the process of the validation of properties of a building element or building elements against the rule requirements, which could vary within a property of a building element such as the height of a door or the area of a space or within a property of a combination of several building elements such as the distance of a route.

Thus, validation of a rule can be denoted by using symbol R for a rule and e for a building element.

Rule validation:

R (e) or R ({e}) = TRUE | FALSE,

where R is a rule, e is a building element, and {e} is a combination of building elements.

Assumption 2*: A circulation rule consists of the logical combination of possibly multiple conditions.*

As shown in Table 3, a circulation rule consists of many explicit or implicit conditions composed of logical connections such as AND, OR and IF~ THEN. The validation of a rule can be evaluated by validating the conditions considering a logical combination of the conditions. The checking of each condition is the process of validating the conditions against a building element or elements. Thus, the validation of a condition can be denoted as a following notation.

R(e) = [logical definitive] C1(e) [logical connection] [logical definitive] C2(e)..

e.g.) R(e) = ! C1(e) AND C2(e) OR C3(e) …. = TRUE | FALSE ,

where,C1, C2… means conditions

and each one can be validated as TRUE or FALSE

logical definitive = NOT (!)| Quantification (ALL, AT LEAST ONE)

Logical connection = AND|OR|IF ~ THEN

Assumption 3*: The validation of each condition entails the checking of a property of a building element that can be classified as a set or derivable by a specified function.*

Each condition works like a function, which has a building element or a combination of building elements as an input parameter, and it returns a validation result of True or False. The internal process of evaluating each condition is associated with the property of the building element(s) to be checked. For instance, the restricted zone condition checks the security zone property of a space. It can be interpreted as if an input building element is a space element; then the element should be included in the restricted zone set.

Thus, a restricted zone condition denoted as C_r(e) can be denoted by using a space element, a space set, or a restricted zone set.

$$C\_r(e) = S\_s(e) \rightarrow S\_re(e), \qquad \text{---------------------------} \quad 1)$$

where $S\_s(e) = \{e|\ e \in SpaceSet\}$, $S\_re(e) = \{e| \in RestrictedZoneSet\}$.

As described above, in order to validate a condition against a building element, the building element should be classified by a required property in the condition. In this case, building elements are classified by two sets: a space set and a restricted zone set. However, some property is difficult to define as a set. For instance, the distance of a route, which is a continuous value, should not be defined using a set because a set is suitable for classifying a discrete property. Thus, if case building elements are difficult to classify by specific properties, then the condition is interpreted using a function, which deals with the properties. For instance, a route condition can be denoted by using a length function, which checks the length of a route against the required length and determines a validation result of True or False.

5.2.3 An abstracted representation of circulation rule validation

The notation described above shows how a rule can be represented with a combination of circulation rules with logical connections and how each condition validation can be validated with sets of building elements classified by its properties or functions.

However, the notation should be expanded so that it represents circulation rule validation because real building design can have many start and target spaces and multiple routes between them. Thus, the cardinality condition regarding the start space, the target space, and the route should be managed for checking. The previous notation is expanded to deal with cardinality issues of circulation rules.

In order to deal with the cardinality issue regarding the start space, the target space, and the route, three indices are defined: the index for a start space, i, the index for a target space, j, and the index of a route, *n*. By using these three indices, a route between a start and a target space is defined as

$$A\ route\ =\ \mathbf{R}^{n}_{Ssi,\ Stj}$$

where n is an index of a route, i is an index of start space Ss and j is an index of target space St.

A route between two terminal spaces is an ordered set of building elements.

For instance, if a route focuses on space elements only, then a n-th route (See Figure 30) between i-th start space Ssi, and j-th target space Stj, which is connected through spaces S1, S2, S3 in that order can be denoted with an ordered set of spaces between Ssi and Stj.

$$R^{n}_{Ssi,Stj}\ (s)\ =\ \{(Ssi,\ S1),(S1,\ S2),(S2,\ S3),(S3,\ Stj)\}$$

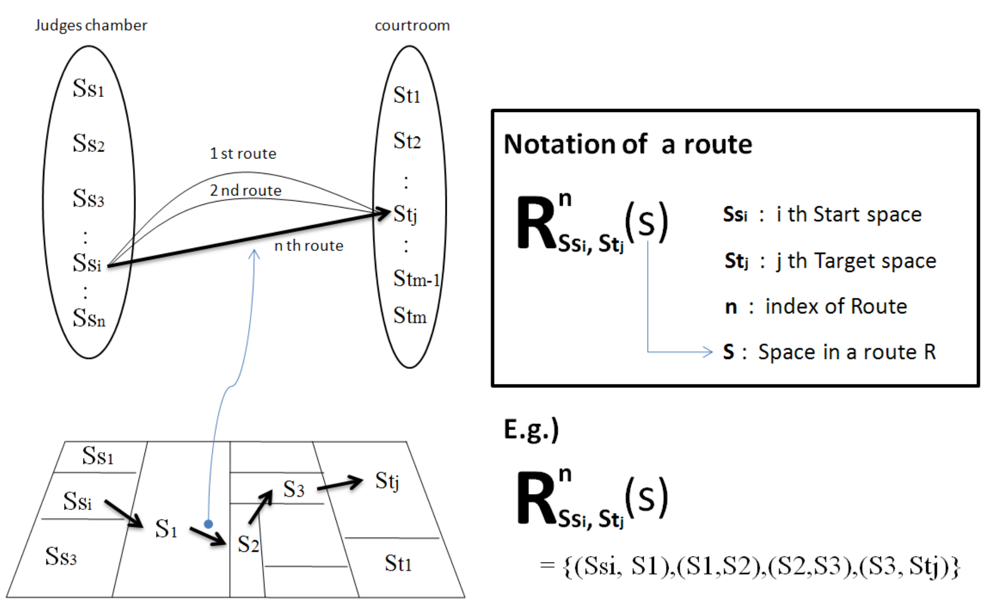


Figure 29 Representation of a route by using an ordered set

By using this notation, the interpretation of each condition can be expanded by specifying a route to which the condition is applied. For example, the notation on restricted zone condition1) can be rewritten as follows.

$R^{n}_{Ssi,Stj}$ *(s)* $\rightarrow$ *S_s(s) AND S_re(s),*

where, S_s(s) = a set of spaces,

S_re(s) = a set of spaces in a restricted zone.

The restricted zone condition can be better developed with cardinality conditions. For instance, the restricted zone condition between two end spaces(start and target space) with one route cardinality condition and one target space cardinality condition can be represented with route index *n* and target space cardinality j.

$\exists j \; \exists n \; | R^{n}_{Ssi,Stj}(s) \rightarrow S\_s(s) \wedge S\_re(s) \wedge \ldots\ldots |$

The required space condition checks the existence of a required space in a route. Thus, it can be interpreted with an element containment notation in a set.

$rs \in R^{n}_{Ssi,Stj}(s)$ *, where rs is a required space.*

The condition can be interpreted as the relationship between building elements in a route and a set of building elements classified according to their properties. However, as described above, for some properties, it is hard to classify building elements such as distance because they are derived after a route is selected, and since the value of distance is a continuous value, it is hard to classify routes by distance. Thus, the following functions are devised to deal with properties that are difficult to classify while checking the parametric conditions defined for occupant circulation rules.

Table 4 Functions of the circulation conditions

Definition of Functions	**Return**	**Purpose**
Size (set) = N	Return number of elements	It is used to determine the size of a set. It can be applied to check a direct access condition by counting the number of transition spaces.
Order (set) = {ordered set}	Return ordered set after ordering of an input set.	It can check the order of input building elements. It can be applied to checking a condition of the order of building elements in a route.
Connection(a building element) = {set of connected building elements}	Return connected building elements as a set	It can check which building elements are connected to which input building element(s).
Length (a route) = length	Length of a route	It can return the distance of an input route. It can also be used for length condition checking.

By using the functions, circulation conditions such as the direct access and the length conditions can be interpreted.

Direct access condition:

$$\text{Size}(R^{n}_{Ssi,Stj}(e)) < 2$$

Length condition:

$$\text{Length}(R^{n}_{Ssi,Stj}(e)) < \text{given value}$$

5.2.4 Application of logical representation to the interpretation of occupant circulation rules in the *U.S. Courts Design Guide*

The notation is applied to the interpretation of the occupant circulation rules in *U.S. Courts Design Guide.* In this case, only the space element is selected as an element for circulation rule checking because most of the circulation rules in the *U.S. Courts Design Guide* are associated with the properties of a space element.

The circulation rules are interpreted according to eleven conditions, and the notation of each parameterized condition is defined in the following table.

Table 5 Notation of parametric conditions

	Start Space Condition		**Target Space Condition**		**Transition Condition**						
Conditions	**Start Space Cardinality**	**Start space**	**Target Space Cardinality**	**Target space**	**Route Cardinality**	**Space Zone**	**Space Usage**	**Required Space**	**Direct Access**	**Vertical Access**	**Distance**
Nota t-ion	**SCC**	**SSC**	**TCC**	**TSC**	**RCC**	**SZC**	**SUC**	**RSC**	**DAC**	**VAC**	**DSC**

In order to check a circulation rule, a start and a target space should be in a building model; then we can check the transition conditions between them. Thus, the logical relationship between terminal space conditions and transition space conditions is an IF-

THEN relationship, and a transition condition is the combination of many conditions with a AND or OR relationship.

Rule = IF (SSC(startspace) and TSC(target space)), THEN (Transition conditions)

where SSC(startspace) is the condition checking existence of a start space in a building, and TSC(target space) is the condition checking existence of a target space in a building.

Each condition can be interpreted with sets of space elements. The spaces in a building are classified by the properties for checking of the circulation rules. The functions are used in Table 4.

Space_ sets = ***{ private_space, circulation_space, transition_space, restricted zone space, public zone space, secure zone space, vertical circulation space*** *}*

$PS(e) = \{e \mid e \in \text{A Set of Private Spaces}\}$

$CS(e) = \{e \mid e \in \text{A Set of Circulation Spaces}\}$

$TS(e) = \{e \mid e \in \text{A Set of Transition Spaces}\}$

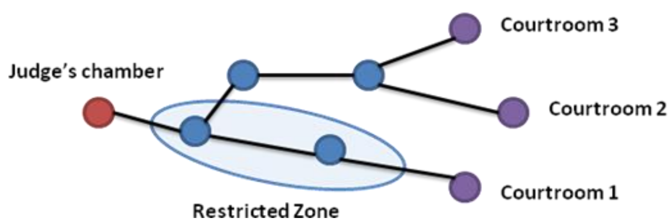
$RzS(e) = \{e \mid e \in \text{A Set of Restricted Zone Spaces}\}$

$PzS(e) = \{e \mid e \in \text{A Set of Public Zone Spaces}\}$

$SzS(e) = \{e \mid e \in \text{A Set of Secure Zone Spaces}\}$

$VS(e) = \{e \mid e \in \text{A Set of Vertical Circulation Spaces}\}$

The circulation conditions are interpreted based on the space sets by space properties. The following examples show how real circulation rules are interpreted.

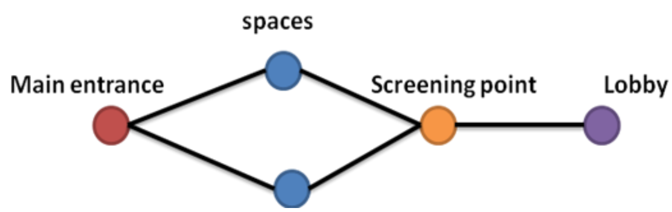
<table>
<tr><td>Rule</td><td colspan="2">Judge's chambers are accessed from restricted circulation with convenient access to the courtrooms.</td></tr>
<tr><td>Interpretation</td><td colspan="2">From the space connectivity viewpoint, the rule regulates the connections between the judge’s chambers and the courtroom as follows:
• The judge’s chambers should be connected to at least one courtroom.
• At least one route between the courtroom and the judge’s chambers should satisfy the following conditions
* All transit spaces in the route should be in the restricted zone.
* All transit spaces in the route should be circulation spaces.
(Because “convenient” is not defined clearly, it is excluded from checking)

</td></tr>
<tr><td>Rule with logical combination of conditions</td><td colspan="2">Rule =
IF(SSC(se) and TSC(te)), THEN (RCC(At least one) and TCC(At least one) and SZC(e) and SUC(e))</td></tr>
<tr><td rowspan="3">Interpretation of each condition with property of a building element</td><td>SSC(se)</td><td>Se ∈Space_Sets</td></tr>
<tr><td>TSC(te)</td><td>te∈Space_Sets</td></tr>
<tr><td>SZC(e) AND SUC(e)[5]</td><td>$\exists j\, \exists n\, |R^{n}_{Ssi,Stj}(e) \rightarrow (CS(e) \wedge RzS(e))|$
where Ss is a set of judge’s chambers. Ssi is an i-th judge’s chamber.
St is a set of courtrooms. Stj is a j-th courtroom.
CS(e) is a set of circulation spaces and
RzS(e) is a set of restricted spaces.</td></tr>
</table>

[5] It means AND relation between space zone condition and space usage condition. See table 5.

<table>
<tr><td>Rule</td><td colspan="2">The public must enter the court building lobby through a single security screening point, controlled by court security officers.</td></tr>
<tr><td>Interpretation</td><td colspan="2">From the space-connectivity viewpoint, the rule regulates connections between the main entrance and the lobby as follows:

- *All lobbies should be connected to the entrance.*
- *All routes between the entrance and lobby should satisfy following conditions:*
** All routes between two spaces should have a screening point.*
(If there is any route without screening point, then it is a problem)

(This case is not limited to the circulation space condition because it requires that all routes be checked, even through private spaces)



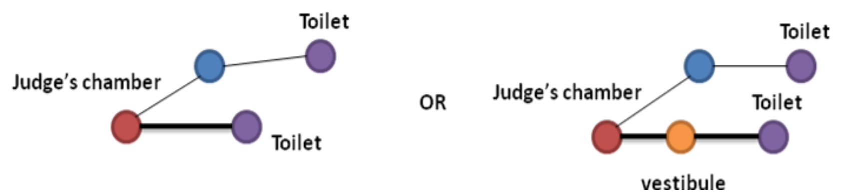
</td></tr>
<tr><td>Rule with a logical combination of conditions</td><td colspan="2">Rule =

IF(SSC(se) and TSC(te)), THEN (RCC(All) and TCC(All) and RSC(e))</td></tr>
<tr><td rowspan="3">Interpretation

of each condition with property of a building element</td><td>SSC(se)</td><td>se ∈Space_Sets</td></tr>
<tr><td>TSC(te)</td><td>te∈Space_Sets</td></tr>
<tr><td>RSC(e)</td><td>$\forall \boldsymbol{j} \forall \boldsymbol{n} \; |R^{n}_{Ssi,Stj}\,\boldsymbol{(e)} \ni \textbf{screening point}\;|$

where Ss is a set of main entrances. Ssi is an i-th main entrance.

St is a set of lobbies. Stj is a j-th lobby.</td></tr>
</table>

<table>
<tr><td>Rule</td><td colspan="2">Judge's chambers should have a directly accessible toilet.</td></tr>
<tr><td>Interpretation</td><td colspan="2">From the space-connectivity viewpoint, the rule regulates connections between the judge's chambers and a toilet as follows:
• The judge's chambers must be connected to at least one toilet.
• At least one route between the judge's chambers and a toilet must satisfy the following conditions:
* All numbers of transit spaces in the route should be zero or one in case the type of transit space is a "transition space" such as a vestibule

</td></tr>
<tr><td>Rule with logical combination of conditions</td><td colspan="2">Rule =

IF(SSC(se) and TSC(te)), THEN (RCC (at least one) and TCC (at least one) and DAC(e) and SUC(e))</td></tr>
<tr><td rowspan="4">Logical representation</td><td>SSC(se)</td><td>se $\in$Space_Sets</td></tr>
<tr><td>TSC(te)</td><td>te$\in$Space_Sets</td></tr>
<tr><td>SUC(e)</td><td>$\exists j\ \exists n\ |R^{n}_{Ssi,Stj}(e) \rightarrow CS(e)|$

where Ss is a set of judge's chambers.

Ssi is an i-th judge's chamber.

St is a set of toilets. Stj is a j-th toilet.

CS(e) is a set of circulation spaces.</td></tr>
<tr><td>DAC(e)</td><td>$\exists j\ \exists n\ |Size(R^{n}_{Ssi,Stj}(e)) == 2$ OR $Size(R^{n}_{Ssi,Stj}(e))=3 \wedge R^{n}_{Ssi,Stj}(e) \ni$ a transition space $|$
where Ss is a set of judge's chambers.
Ssi is an i-th judge's chamber.
St is a set of toilets. Stj is a j-th toilet.</td></tr>
</table>

5.2.5 The algorithm for checking circulation rules

The algorithm for checking a circulation rule is defined in following box by using pseudo code. The process of checking is performed in two steps: Finding a valid route against the conditions (see process 2 in the below box) and repeat the process; and 2) based on the cardinality conditions 6), 7), 8) and 9).

For finding a valid route, the first step is to find a start space and a target space from a building model, 1). At least one start and one target space should be in the building model, or no route would exist.

```
 IF
    Ssi = FindStartSpace (StartSpace)                              ------------------1)
    Tsj = FindTargetSpace (TargetSpace)
THEN
WHILE (Ssi is Exist)
FOR (Each Tsj)
      IsaValidRoute = False;
      AreAllRouteValid = True;

      WHILE (R is Exist)
            R = GetaRouteFromStartSpacetoTargetSpace(Ssi, Tsj );    -----------2)
            Result = Condition.Validation(R);                       --------------------3)
            If(Result == True)                                      --------------------4)
                  IsValidRoute = True;
            If(Result == False)                                     ---------------------5)
                  AreAllRouteValid = False;
            If(Result == True && Tcc= at least one && Rcc = at least one )  -----6)
                  Return True;
```

```
            If(Result == False && Tcc = All && Rcc = All)      ---------------------7)
                  Return False;
            Break WHILE
      END WHILE
      If(IsValidRoute == False && Tcc = All && Rcc = at least one)       --------8)
            Return False;
      If(AreAllRouteValid == True && Tcc = At least one&& Rcc = All) ---------9)
            Return True;
   END FOR
END WHILE


   Variable
   N1 = number of target spaces,
   R = route,
   Tcc = Target space cardinality condition = All| At least one
   Rcc = Route cardinality condition = All| At least one
   FinalResult = Validation result = True | False
   IsValidRoute = Existence of valid route = True| False


   For each start space ssi,
Result = False;
```

If more than one start or target space is found, the next step would be to find a valid route between them against a transition condition. The number of routes and the number of start and target spaces to be validated totally depend on the cardinality condition. If a route or a target space has "at least one" cardinality condition, then it returns true when one valid route is found because the cardinality condition requires just one valid path, 6). If the cardinality condition is all for both a route and a target space, then it returns false when an invalid route is found because the validation is true only when all routes are true for all target spaces, or the validation is false 7).

If the cardinality condition is all for a target space and at least one for a route, then it returns false when there is no valid route for a target space because the cardinality condition requires at least one valid route for each target space, or the validation is false, 8).

If the cardinality condition for target space is at least one, and the cardinality of a route is all, then it returns true if a target space whose routes are all valid routes if found because the cardinality condition requires at least one target space whose routes are all valid.

5.3 An abstracted representation of the checking process with extended logic to cover a model in development

5.3.1 Models in the development stage

The range of diversity of a building design in the developmental stage is open-ended. Thus, the meaning of diversity that must be dealt in this thesis should be defined more clearly. A model in the developmental process can be developed in a diverse way. An addition can be made to an existing building or an existing part of a building can be developed into smaller parts. From a circulation viewpoint, the development of a design in which a new part of a building is added can generate all possible routes, but such routes are difficult to anticipate.

Another way of design development is through the detailing of existing parts of a design. Designing by zoning the main functional parts of a building and refining the zone by adding more specific spaces is a typical design process that architecture building design follows. In this case, the potential route can be generated by compartmentalization, which allows limited expansion because a new route can be generated within a given boundary of a pre-defined space.

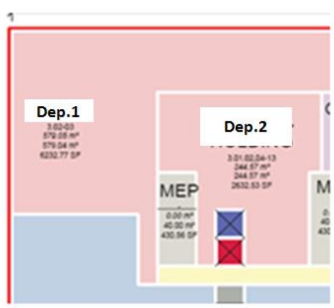
In this thesis, design development generally refers to the process of compartmentalizing spaces in a building instead of adding new spaces to an existing building. Compartmentalization of a space in design development is the division of existing space into sub-spaces. Thus, the sub-spaces are part of an original space. Commonly design development entails zoning a high-level group of spaces such as departments in the early design stage and then developing the zoning area into more detailed sub-spaces within it. Most buildings, including courthouses, are designed according to the concept of departments and individual spaces.

35	District Judge Chambers Suites	-		-
			14	Coat Closet
				Copier Area
				Facsimile Machine Area
				File Storage Area
				Judge Chambers
				Judge Toilet
				Law Clerk Offices
				Reception Area
				Reference/Conference Room
				Secretarial Workstation
				Service Unit
				Storage
				Vestibule
				Visiting District Judge Chambers Suite
	District Judge Courtroom & Associated Spaces	-		-
			21	Clerk Space (Special Proceedings)
				Coat Closet
				Court Reporter/Recorder Office
				Court Reporter/Recorder Storage
				Courtrooms
				Equip. Storage
				Judges Conference / Robing Room
				Judicial Staff Toilets
				Juror Toilets

Figure 31 Aggregated space names in a courthouse

The *U.S. Courts Design Guide* includes a list of departments and individual spaces within each (see Figure 31). Thus, the design of a courthouse can be developed using a department-level space name in the early stage of development, and it can be refined by defining individual spaces in the design.

This development includes the adding of more connections of spaces by adding space-connection building elements such as doors or openings. A review of several courthouse building designs under development found diversity in the modelling of space connections. Some courthouses have only doors for explicitly defining the connection of spaces, and some have no doors (e.g., see Figure 32). Actually, if the model is still in development, it is difficult to determine whether the building has all connections or not because the model in development has the potential to have additional space connections. Thus, from the perspective of space connections, the building model can be classified into two types: a model without connections and a model with connections.



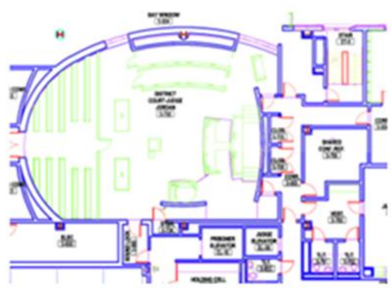


Figure 32 Models without doors and with doors

Another aspect of design development is the specialization of the usage of a space. For instance, a space labelled "clerk's office" can be defined as "District law clerk's office" later. Assigning of a more detailed function to a given space commonly occurs throughout the development of a design. Many other examples such as a "toilet" that later become a "man's toilet" were found in real courthouse design.

These three types of design development—compartmentalizing, adding more connections of spaces, and specializing space usage—are the main factors of design development that this thesis deals with.

5.3.2 Validation based on the pre-defined classification of spaces

In the checking of the late concept model, the validation of each condition returns only two results: TRUE and FALSE. It returns true when a building element satisfies the required condition; otherwise, it returns false. This check is based on the assumption that building elements, especially spaces for circulation rule checking, are developed in detail enough to apply circulation conditions to the space, and its property is identifiable for checking.

Type	Classification
*stair	circulation
stair*	circulation
*vestibule	circulation
vestibule*	circulation
*Reception	circulation
Reception*	circulation
*Secretary	circulation
secretary*	circulation
*corridor	circulation
corridor*	circulation
*elevator	circulation
elevator*	circulation
*sally port	circulation
sally port*	circulation
*sallyport	circulation

Figure 33 Classification of circulation spaces

For instance, in order to check the circulation space conditions of a space, we require a predefined classification of spaces by their use (e.g., see Figure 33), and through this classification, the circulation condition of a space in a building can be checked. This classification-information can be saved as a property attached to each instance of a space in a building model, or it can be formed as an external file for reference during the checking process.

From this classification, spaces in a building are classified into two types: spaces for circulation (circulation-space), and spaces for private usage (non-circulation space). If a space is in the circulation-space set, then it is true for circulation condition checking; otherwise, it is false (see Figure 34).

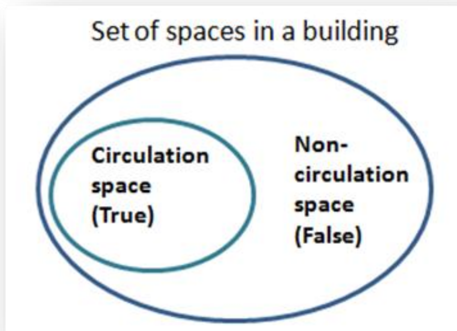


Figure 34 Classification of spaces by the circulation condition

However, a building element in development tends to be too vague to evaluate because it is typically an incomplete model; thus, the information in it could be insufficient for checking.

In order to deal with the incomplete model, this thesis adopts a pre-defined table with a space name and a property classification for circulation rule checking, which is referred to as the Space Classification Table (SCT). The SCT contains the following information.

- **Space hierarchy information**

The space hierarchy concept such as a department and its elementary spaces is a common concept for many types of buildings. In courthouse design, the SCT contains the relationship between a department name and its elementary space names (see Figure 35).

- **Space classification information**

Spaces in a building can be classified in many ways according to their properties. The SCT has classification information of spaces in a courthouse for circulation rule checking such as security level and space usage (see Figure 35).

Department Name	elementary space name	Security	Space usage
District Judge Chambers Suites	Law Clerk Offices	restricted	Non-circulation
	Coat Closet	restricted	Non-circulation
	Copier Area	restricted	Non-circulation
	Facsimile Machine Area	restricted	Non-circulation
	File Storage Area	restricted	Non-circulation
	Judge Chambers	restricted	Non-circulation
	Judge Toilet	restricted	Non-circulation
	Vestibule	restricted	circulation
	Reception Area	restricted	circulation

Figure 35 Space usage classification table

Validation based on this table could have four cases. Let us assume that 'C' means a circulation condition, and we try to validate the spaces consisting of the judge's chamber, the vestibule, and the district judge's chamber suite against the circulation condition by using the SCT. Validation of the judge's chamber is false because it is non-circulation space. Validation of the vestibule is true because it is circulation space. However, validation of the district judge's chamber suites could be either true or false because some part of this area could be developed as a circulation space such as the vestibule, and the other part could be developed as a non-circulation area. In this case, validation could be potentially true or false. Of course, a space could be outside of the pre-defined space name set. If architects use ad-hoc names for unclearly-defined spaces, validation of the spaces cannot be performed because the names of the spaces are not identified well enough for evaluation.

Based on the SCT, the four cases described above are typical cases that could occur when a space is checked during development.

Figure 36 shows the change in classification using set-based notation for covering the four cases in the development-level model. The previous checking methods—the graph-based checking method and the set-based checking method—classify spaces with true and false. However, the generic method uses four types of classifications to deal with the four cases described above: true, false, potential (true or false), and unidentified.

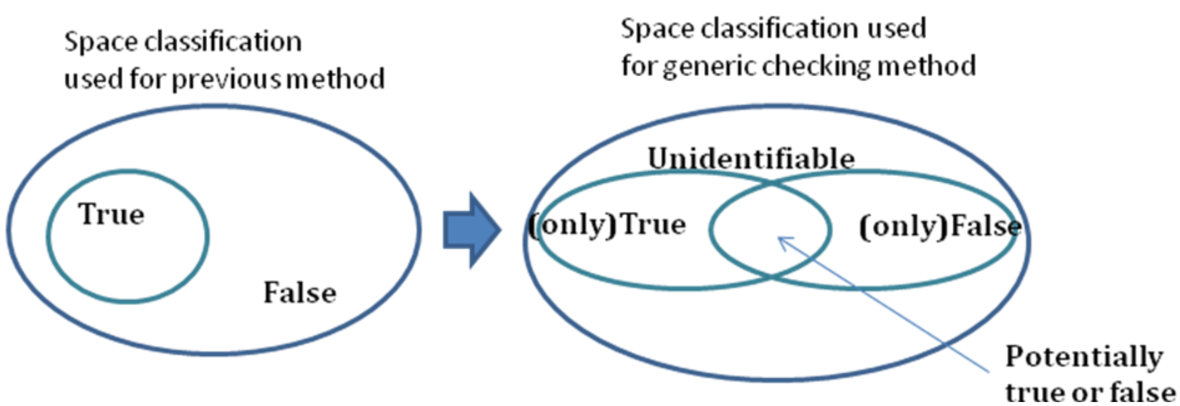


Figure 36 Spaces in the four types of classification

For instance, by using these terms, the validation of the four spaces (i.e., the judge's chamber, the vestibule, the district court judge's chamber, and the non-identifiable space) using the SCT in Figure 35 could be described as follows:

C(Judge Chamber) = FALSE

C(Vestibule) = TRUE

C(District Judge's Chamber) = POTENTIAL TRUE

C(non-identifiable space) = UNIDENTIFIABLE

Thus, the four types of validation—true, false, potential true (= potential false), and unidentifiable—are defined as follows:

TRUE

If a space is in a classification that satisfies the required condition, it returns a true validation.

FALSE

If a space is in a classification that does not satisfy the required condition, it returns a false validation.

POTENTIAL TRUE (= POTENTIAL FALSE)

If a space could belong to either a true or false classification through development, it returns a potential true validation.

UNIDENTIFIABLE

If a space is not identifiable in the SCT, it returns a non-identifiable validation.

Therefore, the validation of a building element in development must fall into one of these four types. However, the validation of a building element in the final stage of a design falls into only one of two types: true or false.

C(e) = true/false,

where e is an element in the final stage design.

C(e) = true(T)/false(F)/potential(P)/ Unidentifiable (U),

where e is an element in development.

5.3.3 Validation based on the type of space in the development stage

After a review of several building models—the courthouse 2, courthouse 5 models—space elements associated with circulation checking could be classified into three main types: individual spaces, aggregated spaces, and general spaces.

An individual space is a space that is developed fully enough to check it against the circulation rules. It can be mapped directly onto the circulation rules. For example, a district court judge's chamber can be mapped to a start space in the rule "The district court judge's chamber should be connected to a district courtroom through a restricted zone." An aggregated space is a space in which individual spaces are aggregated. For example, a district court judge's chamber suite—the district courtroom related offices—are examples of an aggregated space. Many building models in development have aggregated spaces that can be further developed into individual spaces. A general space is a space that has a general name instead of a specified name such as "courtroom" instead of "district courtroom," "bankruptcy courtroom," or "magistrate courtroom." Many building models under development have such general spaces because the architect(s) have not decided on specific name.

From the set perspective, an aggregated space and a general space are both sets that have individual spaces as elements. For example, the judge's chamber suites and courtroom can be written as a set.

Judge's chamber suites =

{judge's chamber, law clerk's office, secretary office, judge's library, judge's toilet …}

Courtroom =

{District courtroom, Senior District Courtroom, Bankruptcy courtroom …}

If a space element in a building model has its property information required for explicit rule checking, then the type of the space element is not an important matter. We can validate the space element with the existing properties. A problem arises when a space element does not have required information that must then be derived from existing data. We assume that individual space elements are classifiable according to their properties, but building models also consist of aggregated or general elements. Thus, we have to guess the classification of the aggregated or general elements by using the given classification of individual elements.

Aggregated space and general space elements are a group of space elements, so this thesis refers to them as a "space group." From the perspective of the properties of a space element, the space group could have three cases (see Figure 37); the space group totally belongs to the (A) conditions, the group partly belongs to the (B) conditions, and the group does not belong to the (C) condition.

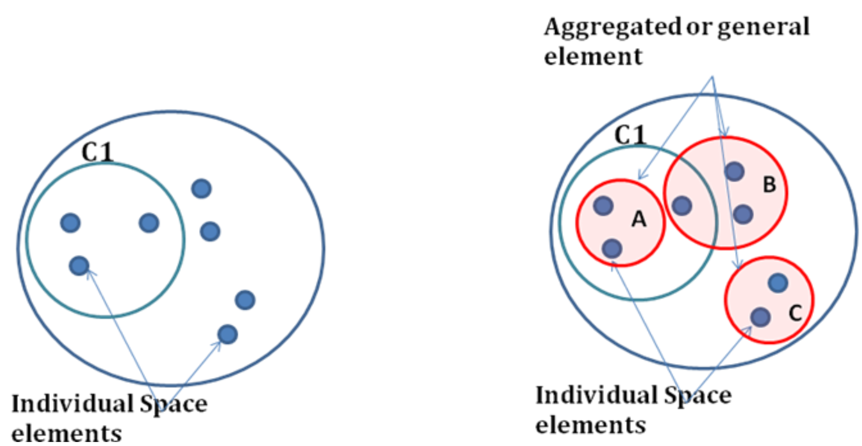


Figure 37 Individual, aggregated, and general spaces in a building

Validation of each condition with a group of elements depends on the type of cases of the three cases. Case A returns a validation of true because all the sub-elements in group A are in condition C1.

Case C returns a validation of false because all the sub-elements in group C are in condition C1.

However, Case B could also be true or false according to elements in set B. It can be denoted by using logical symbols.

Space group: sg = {se| se ∈ a set of space elements}

C1(sg) = TRUE, when $\forall$se | sg(se) → C1(se)| == TRUE ------------1)

C1(sg) = FALSE, when $\forall$se | sg(se) → C1(se)| == FALSE ------------2)

C1(sg) = UNIDEN , when $\exists$se | sg(se) → C1(se)| == TRUE ------------3)

AND $\exists$se | sg(se) → C1(se)| == FALSE

5.3.4 Three levels of validation for the transition condition

The previous section shows how a space can be validated by a condition of true, false, potential true, or unidentifiable. However, real validation of a circulation rule requires a combination of validations of many spaces against multiple transition conditions. In order to validate a route, all individual spaces in a route should be evaluted against multiple conditions of the circulation rules. This process of validation consists of three steps: a condition-level validation, a transition-level condition, and finally a route-level validation.

Condition-level validation

This level indicates the validation of a space by a condition as described above (e.g, see number 1 in Figure 38).

Transition-level validation

This level of validation involves checking a space against a transition condition, which is a logical combination of multiple conditions. Thus, the final validation of a space against a transition condition is decided by logical operations of the validation results of each condition(e.g, see number 2 in Figure 38).

Route-level validation

This level is the combination of the validations of spaces against a transition condition. A route consists of many transition spaces, each of which is validated against a transition condition. A route-level validation is the logical combination of the validation results of the transition condition(e.g, see number 3 in Figure 38).

For instance, in the route between a space, S1, and a space, Sn, A, B, and C are transition spaces(see Figure 38). In this case, validation of this route should be done on all three levels of validation. Let's assume that the transition spaces A, B, and C are classified against each condition C1 and C2, as in Table 6, and the transition conditions of Rule X comprise two conditions, C1 and C2, and the two conditions have an AND relationship (see Table 7).

Table 6 Space validation table based on classifications C1 and C2

Space	Classification	
	C1	C2
A	True (T)	Potential True (P)
B	True (T)	False (F)
C	Unidentifiable (U)	True (T)

Table 7 Example rule X with conditions C1 and C2

Rule X				
Start Space	Transition Condition			Target Space
	Condition	Relationship	Condition	
S1	C1	AND	C2	Sn

According to this classification table, a condition-level validation for space A against each condition C1 and C2 is T and P in each(see number 1 in Figure 38).

Transition condition-level validation is the final result of the combination of each condition's validationof the transition conditions. If T and P are the validation result of each condition, the transition condition-level validaton result is logical–an AND combination of T and P. Currently, the local AND combination of T and P is defined P (this definition is explained in detail in the next section). Thus, the transition condition level validaton result for space A is P (see number 2 in Figure 38). The logical combination of transition condition level validation results for each space A, B and C is a route level validation (see number 3 in Figure 38) . In the case of Figure 38, the results of transition condition level validation are P, F and U for each space, A, B and C. thus, the local combination of them, which is a route level valdiation is F.

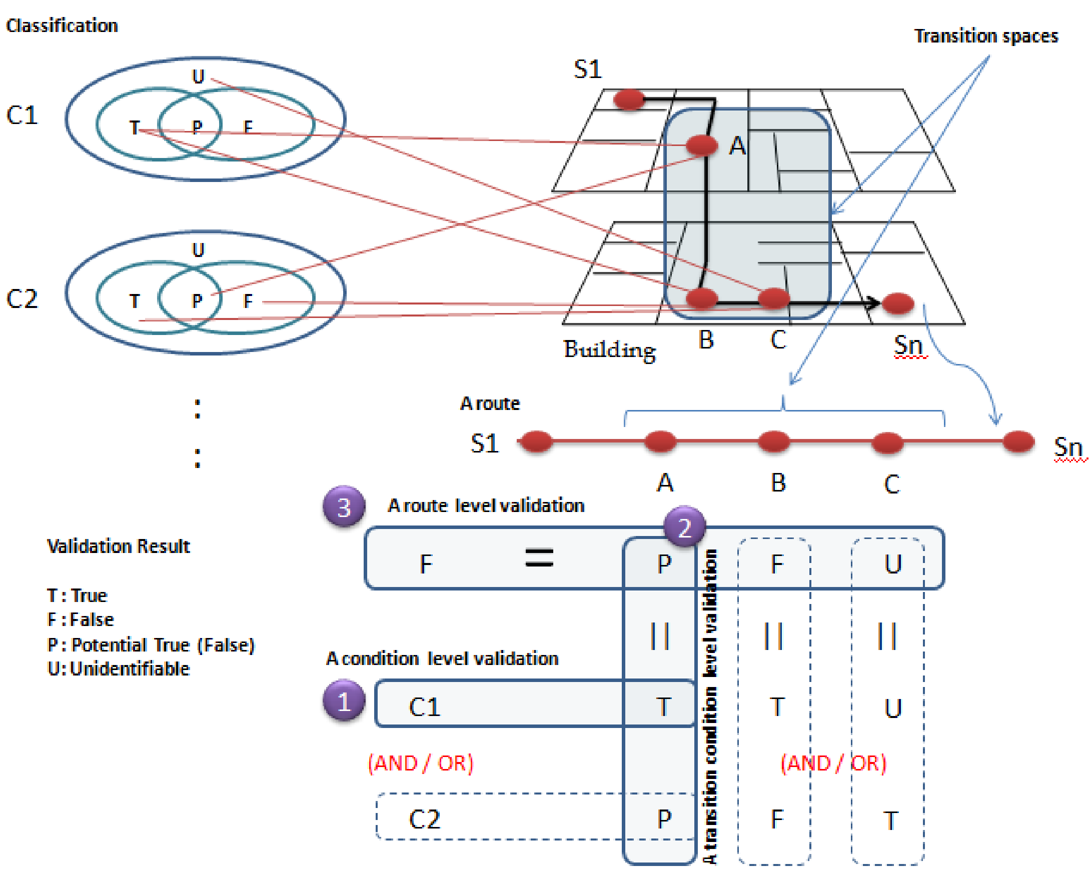


Figure 38 Three levels of validation

- **Transition condition-level validation**

A transition condition is a logical combination of circulation conditions. Thus, the validation of a transition condition is the logical combination of the validation of each circulation condition. In order to deal with the logical combination of the violation results in a development model such as true (T)/false (F)/potential(P) and unidentifiable(U), logical connectors(AND and OR) are defined.

If we say TRUE is positive and FALSE is negative, the logical operator AND returns a negative validation result and OR a positive validation result. For instance, a statement including F with an AND connector returns F because AND returns mostly negative validation results in the statement, and a statement with T with an OR connector returns T because OR returns mostly positive ones. The same concept is applied to defining the logical connection for validation results (T, F, U, and P). First, I decided that an unidentifiable (U) case is more negative than a potential (P) case because the potential case indicates it has a chance to be true even if it also has a chance to be false. However, in the U case, we do not know even if it has a chance to be true or not. Therefore, the order of the validation results—T, P, U, and F—is defined as follows.

(Positive) T > P >U> F (negative)

The AND operator returns more negative validation results and the OR more positive results. Thus, we have following logic table of AND and OR connectors for the validation results of T, P, U, and F (see Table 8).

Table 8 The AND, OR operators for circulation rule checking

T AND T = T	T OR T = T
T AND F = F	T OR F = T
T AND P = P	T OR P = T
T AND U = U	T OR U = T
F AND P = F	F OR P = P
F AND U = F	F OR U = U
P AND U = U	P OR U = P

The validation of a building element (e) against a transition condition is the combination of each validation of a condition.

A transition condition (TC) = C1 [AND/OR] C2 [AND/OR] C3 …..

A transition condition (e) = TC(e)

= C1(e) [AND/OR] C2(e) [AND/OR] C3(e) …..

= True | False | Potential | Unidentifiable

- **Transition route-level validation**

A route is an ordered set of building elements. Thus, the validation of a route against a transition condition indicates the validation of all the building elements against a transition condition. As described above, each validation of a building element against a transition condition could be T, F, P, or U, or any combination of them. The possible cases of the combination of these validations can be calculated by the sum of following four cases.

The first case is that a route consists of only one type of validation such as T only, F only, P only and I only; the second case is that a route is comprised of two types of these four types of validation; the third case is that a route consists of three of the four types of validation; and the final case is that a route comprises all four types of validation. Each case can be calculated by the following formula.

$$_4C_{1.} + {_4C_{2.}} + {_4C_{3.}} + {_4C_{4.}} = 2^4 - 1 = 15 \text{ cases.}$$

Each case of a combination is validated as Table 9 shows. The validation of a route is true only when all the building elements in a route are validated as true. If there is at least one building element validated as false, then the validation of the route is false. A route is potential if the route consists of only potentially valid building elements or a combination of only truly validated building elements. An unidentifiable case is when a route does not have a false-validated building element but it has at least one unidentifiable building element.

Table 9 Possible cases in route validation

Route Validation	$_4C_{1.}$	$_4C_{2.}$	$_4C_{3.}$	$_4C_4$
True	T only			
False	F only	(F, T), (F,P), (F,U)	(F,T,P),(F,P,U),(F,T,U)	(F,U,P,T)
Potential	P only	(P,T)		
Unidentifiable	U only	(U,T), (U,P)	(U,P,F)	

As a result, the validation of a route follows the most negative validation in a combination of validation results. Thus, briefly, it could be said that all elements in a route have an AND relationship for validation.

A route- R = {{e1, e2}, {e2,e3}, {e3,e4} …. }

= (e1, e2, e3, …….)

A transition condition (R) = TC(R)

= TC(e1) AND TC(e2) AND TC(e3) ……..

5.3.5 Validation with a terminal condition and a transition condition

The main structure of an occupant circulation rule is defined by using an IF~THEN relationship between a terminal space condition and a transition condition.

The fundamental structure of circulation rule

IF(Start and target space condition), THEN (transition condition)

The existence of a start and a target space are the pre-conditions for checking the transition condition between them. Thus, a final validation should be made by checking the terminal condition. Theoretically, terminal condition checking could also have four results: true, potential, unidentifiable, and fail. However, in the process of the identification of terminal spaces, fail is the same as unidentifiable. Thus, unidentifiable and fail can be considered the same case.

True

If the terminal space is identified in a model, then it returns true.

Potential

A terminal space could be an aggregation space or a general space of the terminal space to be checked. For instance, the judge's chamber suite for checking the judge's chamber could be an example of a case of an aggregation space for a terminal space.

Unidentifiable (fail)

If we cannot locate a terminal space in a building model, then it is hard to determine whether the model has one or not or whether it is identifiable or not because of misspelling of the space name or a different name altogether. Thus, this thesis regards this unidentifiable case as an identification issue instead of a missing space, and it validates this case as “unidentifiable.”

The following is the table of the validation of all possible cases. If the validation of the “If condition” is true, then the final validation is the same as the validation of the “Then condition.” If the validation of the “If condition” is U(F), then we do not need to check the “Then condition.” Thus, the final validation is “No checking.” If the validation of the “If condition” is P, then all the validation results of the “Then-condition” have the potential to be true. For instance, if the “Then condition” is true, it means that it is potential true instead of true. Thus, a final validation is made by using the combination of the two results (see Table 10).

Table 10 Validation of the IF-THEN condition

IF (Start and Target Space Condition)	THEN (Transition Condition)	Final Validation
T	T	T
	P	P
	U	U
	F	F
P	T	P-T(potential- true)
	P	P-P(potential-potential)
	U	P-U(potential-unidentifiable)
	F	P-F(potential-fail)
U(F)	No checking	No checking

5.4 Integrated checking with adjacency and connection graph-based checking

A building design can also be developed by adding more connections between adjacent spaces. In order to deal with potential connections, the generic method adopts two types of graphs: a ***connectivity graph*** and an ***adjacency graph***. The former represents the connectivity of spaces through building elements that are connecting spaces such as doors, openings, stairs, and elevators. The latter represents the adjacency of spaces instead of real connections between them. This thesis adopts an adjacency graph to represent the potential connections of spaces in a model being developed and a connectivity graph to represent the connections of spaces existing in a current model (see Figure 39).

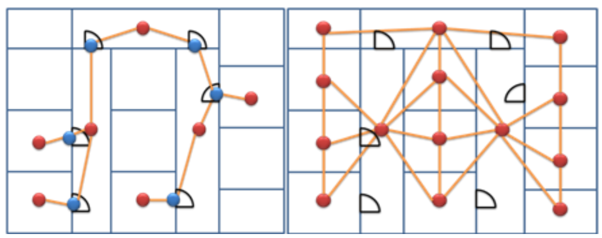


Figure 39 Connectivity and adjacency graphs

The main difference between the new approach and the previous approach is the use of these two graphs for checking. Although checking with a connectivity graph reflects the current status of a building design against a rule, validation with this type of graph may not be the final result because the design is still in development. An adjacency graph can show the potential validation of a design because it contains all potential routes in a building. Thus, the final routes in a building design are included in the adjacency graph, and the routes in the connectivity graph of a current building design could be a part of the final routes. The relationship among these three set of routes can be represented by using a set (see Figure 40).

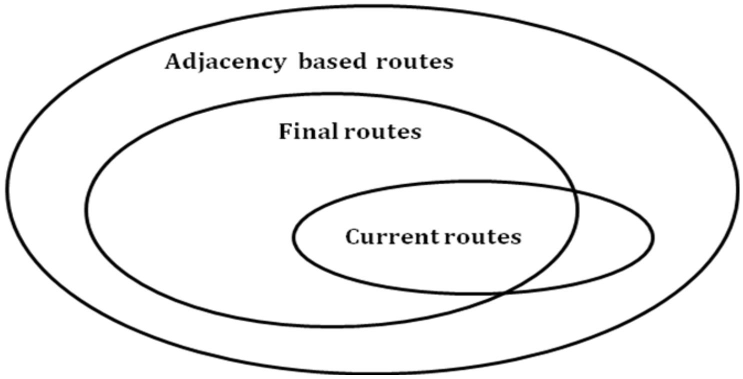


Figure 40 The relationship among the routes in the three graphs

$$\textbf{\textit{Routes in Adjacency Graph}} \supseteq \textbf{\textit{Routes in Connectivity Graph}}$$

The checking of rules based on the two graphs (i.e., the adjacency and connectivity graphs) can provide more information about the current building design than the checking of rules with only one graph. The set-based method of checking is based on the grouping of spaces based on adjacency. Thus, set-based checking is partial checking based on an adjacency graph. Set-based checking returns true if two spaces have an adjacency relationship in a given condition. However, it does not consider the existence of doors because it totally focuses on the model in its early stage when it has no doors. Thus, validation by set-based checking is same as that by adjacency-based checking (see Figure 41).

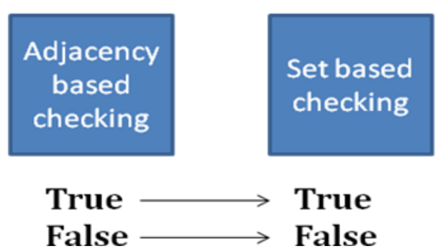


Figure 41 Validation by adjacency and set-based checking

However, a model in the development stage can have more information on space connectivity even though it contains only a portion of the final design connections. Current connection information can give a more in-depth view of the results from adjacency-based checking. For instance, both true validation by adjacency graph-based checking and current graph-based checking means that at least one actual route between spaces must be checked and that it satisfies the required conditions in a given rule. Thus, if a rule requires at least one route condition, it is a really true.

However, validation could be true in an adjacency graph but false in a current connection graph. In such a case, the building design has the potential to be true, but it is false in the current stage. Set-based checking regards this case as true, but checking with the current connection graph can distinguish between a real true and a potential true.

The same approach is applied to checking a model in the development stage, particularly if the model in the development stage does not have enough information to check some rules. As described below, the spaces in a development level model could have four types of validation results. Both an adjacency graph and a connectivity graph can have four types of validation results. Thus, a possible combination of validation results from the connection and adjacency graphs will be more diverse, and an interpretation of the checking results from the connection and adjacency graphs must undergo a more careful review so as to have a more valid conclusion from checking. Chapter 6 reviews all the possible cases of checking both the adjacency and connectivity graphs and discussed how the case can be interpreted reasonably.

Chapter 5 addresses many topics in separate sections. Each of these sections should relate to the main objectives of this thesis, which are discussed in the summary.

5.5 Summary

This section assumes that rule validation is a process of checking of properties of building elements against the conditions in a rule with logical connections. In particular, this thesis assumes that circulation rule validation is a process of checking routes between start and target spaces with consideration of cardinality conditions. Notations of a route between the start and target spaces, defined using an ordered set of spaces in a route, show how the validation of a route can be interpreted by using a set of ordered spaces. This form of representation is developed to support the diverse level of the modelling of spaces in the development stage of the design process. The representation deals with individual spaces, aggregated spaces, and generalized spaces. This section describes how these diverse types of spaces can be evaluated according to three levels: the condition level, the transition condition level, and the route level.

The next section explores how this validation process can be applied to the checking of ten parametric conditions in the occupant circulation rules.

CHAPTER 6

DEVELOPING THE APPLICATION OF A GENERIC CHECKING METHOD TO COURTHOUSE DESIGN

6.1 Application of the basic concept to circulation rule validation

This chapter applies the generic checking method to the validation of circulation conditions with a sample courthouse plan (see Figure 42). The likelihood of checking a certain condition depends on the existence of required information in the model regardless of the level of completeness of a model. Even if a space in a building is only roughly defined, it can be validated if the space has information that can be checked as additional information about the space such as the P-set in IFC. However, a model in the development stage does not always have the required information in some additional form, so its validation might be unidentifiable because of the lack of information. This section reviews, in detail, the possible cases according to the type of condition and the way of modeling.

- **Validation of the restricted zone condition**

Let's assume that the restricted zone conditions in the following rule are checked in three cases in a Courthouse plan (see Figure 42).

Rule: The USDC judge's chamber should be connected to the library through the restricted condition.

CASE 1: Space b is a corridor

CASE 2: Space b is a building support area

CASE 3: Space b is a district judge's chamber suite

The courthouse below has a connection between the USDC judge's chamber and the library through spaces a, b, and c. Now if the route though spaces a, b, and c is assumed the only route among them, then validation of the rule totally depends on the type of connection spaces. If the connection spaces are all restricted corridors, then the validation is true. However, there could be a variety of connection spaces in the development level mode

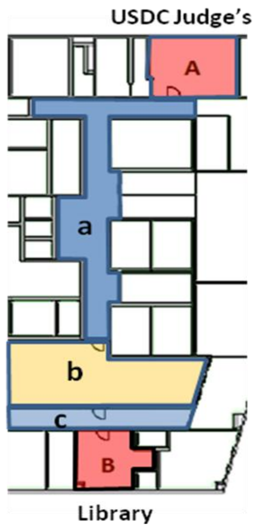


Figure 42 A courthouse plan

If space b is a corridor (CASE 1), it could be specified later as a restricted, public, or secure corridor because a courthouse has these three types of corridors. Thus, it has a potential to be validated as true, but it is unidentifiable in this stage. If space b is a district building support area (CASE 2), it could be developed into many sub-spaces in the department. The relationship between a district court judge's chamber and its sub-spaces is one of a part and an aggregation. If an aggregation includes a part that satisfies a required condition, then it has a potential to be true. In this case, some of the spaces in the building support area are in a restricted zone and some are not. Thus, it has a potential to be validated as true, but it cannot be decided.

These two cases (CASES 1 and 2) are both unidentifiable case 3) because the space elements in the space groups (corridor, building support area) partly belong to the restricted zone. However, in CASE 3, the validation is true because all the subspaces in the restricted judge's chamber are in the restricted space.

As a summary, the following three types are possible based on this classification.

Type 1: A space in model is a group space (an aggregated space or a general space), and all its sub-spaces are in a restricted zone.

The validation is always true because the development of a space into a sub-space is bounded to a restricted space.

Type 2: A space in a model is a group space (an aggregated space or a general space), and some of its sub-spaces are in a restricted zone.

The validation is unidentifiable because the development of a space into a sub-space could become a non-restricted space.

Type 3: A space in a model is a group space (an aggregated space or a general space), and none of its sub-spaces are in the restricted zone.

The validation is always false because the development of the space to a sub-space could not be a restricted space.

Table 11 Validation table of the zone condition

Space Type in Model	Sub-space		Validation of Zone Condition	
			Adjacency graph	Connection graph
A Group Space (A general space or an aggregated space)	Type 1	All sub-spaces are in the required zone	True	True
	Type 2	Only some of the sub-spaces are in the required zone	Unidentifiable	Unidentifiable
	Type 3	No sub-spaces are in the required zone	False	False

The validation of these three cases can be applied to both connection and adjacency graphs because the zone condition considers only the zone property of spaces instead of the connectivity of spaces. The validation of these cases for the restricted zone condition is summarized in Table 11.

- **Validation of the circulation condition**

Routes in the adjacency graph are potential instead of real connections. Thus, the validation of a circulation condition in an adjacency graph differs from that in a connectivity graph. An adjacency graph returns true if the route has potential connectivity, but a connection graph returns true only when there is a real connection. Thus, an adjacency graph assumes that two spaces are accessible if the two spaces are adjacent. It considers all sub-spaces in an aggregated space accessible to each other regardless of the way of they are compartmentalized in the aggregated space. Thus, the compartmentalization of an aggregated space in adjacency graph-based checking does not affect the validation. In Figure 43, spaces a and c have an adjacency relationship regardless of how space c is compartmentalized.

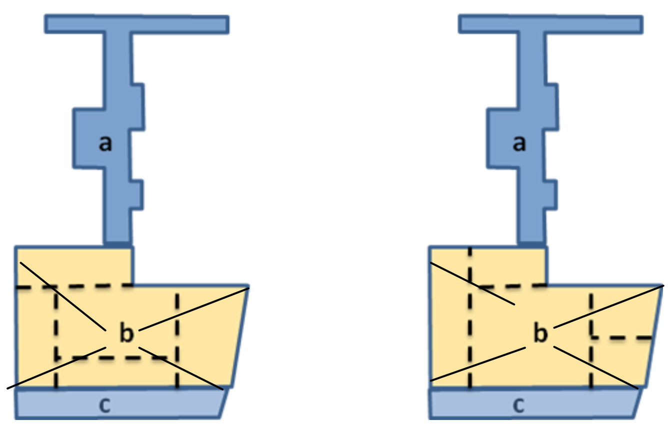


Figure 43 Compartmentalization of a general space

However, the way of compartmentalizing affects the connectivity of two spaces. From the connectivity viewpoint, internal connectivity in an aggregated space is always unidentifiable because of the variety of ways of compartmentalization and the location of doors in a compartment. Based on this assumption, six cases in the development level model are investigated.

CASE 1: A space in the model is a general space, and all of its sub-spaces comprise a circulation space.

The validation of this case is always true in both the adjacency and connectivity graphs because the design development, which is the selection of one of the sub-spaces, is a circulation space.

CASE 2: A space in the model is a general space, and some of its sub-spaces comprise a circulation space.

The validation of this case is unidentifiable because the development of a general space to one of its sub-spaces could not be a circulation space in both the adjacency and connectivity graphs.

CASE 3: A space in the model is a general space, and none of its sub-spaces comprise a circulation space.

The validation of this case is always false because the development of a general space to one of its sub-spaces is not a circulation space in either the adjacency or the connectivity graph.

CASE 4: A space in the model is an aggregated space, and all its sub-spaces comprise a circulation space.

The validation of this case is always true in the adjacency graph because the development of the aggregated space by compartmentalizing it to its sub-space will become a set of circulation spaces that are in an adjacent relationship. However, validation is unidentifiable for the connectivity graph because the connection of sub-spaces totally depends on how they are compartmentalized and where doors are located.

CASE 5: A space in the model is an aggregated space, and some of its sub-spaces comprise a circulation space.

The validation of this case is unidentifiable in both the adjacency and connectivity graphs because the development of an aggregation space cannot guarantee either the adjacency or connectivity of spaces only through a circulation space.

CASE 6: A space in the model is an aggregated space, and none of its sub-spaces comprise a circulation space.

The validation of this case is always false in both the adjacency and connection graphs because the development of an aggregated space comprises a non-circulation space.

Validation of the six cases is summarized in a following table.

Table 12 Validation table of the circulation conditions

Space Type in Model	Sub space		Validation of Circulation condition	
			Adjacency graph	Connection Graph
General Space (Kind of)	Case 1	All sub-spaces comprise a circulation space	True	True
	Case 2	Only some of the sub-spaces comprise a circulation space	Unidentifiable	Unidentifiable
	Case 3	No sub-spaces comprise a circulation space	False	False
Aggregated Space (Part of)	Case 4	All sub-spaces comprise a circulation space	True	Unidentifiable
	Case 5	Only come of the sub-spaces comprise a circulation space	Unidentifiable	Unidentifiable
	Case 6	No sub-spaces comprise a circulation space	False	False

- **Validation of the direct access condition**

Validation of the direct access condition can typically be checked by counting the number of transition spaces between two terminal spaces. An adjacency-based graph can also count the number of transition spaces even if a connection is only potential. The following are the possible cases for direct access condition checking.

Note: The number of transition spaces below refers to the number of transition spaces in the shortest route.

Case 1: The number of transition spaces is more than one.

The validation of the direct access condition is always false in both the adjacency and connectivity graphs because no shortest route directly connects the transition spaces.

Case 2: The number of transition spaces is one, and the transition space is an entrance space such as a vestibule.

The validation is always true for the connectivity graph but unidentifiable for the adjacency graph because no route can be made through the shortest adjacency connection.

Case 3: The number of transition spaces is one, and the transition space is not an entrance space.

The validation is always false in both the adjacency and connectivity graphs because the connection is not through an entrance space.

Case 4: The number of transition spaces is zero

The validation is true for the connection graph but unidentifiable for the adjacency graph because direct adjacency does not guarantee the design development through it.

Validation of these four cases is summarized in the following table.

Table 13 Validation table of the direct access condition

The number of transition space in shortest route	Detailed case	Validation of the direct access condition	
		Adjacency graph	Connection graph
More than one	Case 1	False	False
One	Case 2 If the transition space is an entrance space such as a vestibule or an entrance	Unidentifiable	True
	Case 3 If the transition space is not an entrance space such as a vestibule or an entrance	False	False
zero	Case 4	Unidentifiable	True

- **Validation of the required space condition**

Checking the required space condition is straightforward because it checks the existence of required conditions in the transition spaces. If the transition spaces include general or aggregated spaces, then the checking should consider their sub-spaces. Required space checking can be classified into three main cases: when the transition space does not include a general or an aggregation space, when a transition space includes a general space, and when a transition space includes an aggregation space.

-When a transition space does not include a general space:

Case 1: A required space is in a transition space.

The validation is always true in both the adjacency and connectivity graphs because the tested route has a required space.

Case 2: A required space is not in a transition space.

The validation is always false in both the adjacency and connectivity graphs because the tested route does not have a required space.

-When transition spaces include a general space:

Case 3: A required space is in a general space.

The validation is unidentifiable in both the adjacency and connectivity graphs because the general space could be a required space.

Case 4: A required space is not in a general space.

The validation is false in both the adjacency and connectivity graphs because the general space does not have a required space.

-When transition spaces include an aggregation space:

Case 5: A required space is in an aggregated space.

The validation is true for the adjacency graph because an adjacency route has a required space. However, the validation is unidentifiable for the connectivity graph because the connection of spaces in the aggregation space could not have a required space.

Case 6: A required space is not in an aggregated space.

The validation is false in both the adjacency and connectivity graphs because no route in the aggregated space can have a required space.

The validation of these six cases is summarized in the following table.

Table 14 Validation table of the required space condition

Transition spaces		Validation of required space condition	
		Adjacency graph	Connectivity graph
No general or aggregation space in transition spaces	CASE 1 : Transition spaces include a required space	True	True
	CASE 2 : Transition space does not include a required space	false	false
General space as a transition space	CASE 3: Subspaces of the general space have a required space	unidentifiable	unidentifiable
	CASE 4: Subspaces of the general space do not have a required space	false	False
Aggregation space as a transition space	CASE 5: Subspaces of the aggregated space include a required space	true	unidentifiable
	CASE 6: Subspaces of the aggregated space do not include a required space	false	false

6.2 Final validation with the integration of the results from the adjacency and connectivity graphs

Final validation is done by the integration of both the results of the connectivity graph (CG)- based checking and adjacency graph (AG)-based checking. CG-based checking can have four types of validation results: true, potential, unidentifiable, and false.

CG-based checking returns a true validation result if a route satisfies all the required conditions (A) but it returns a false if no correct route satisfies the transition conditions

(B). In addition, CG-based checking returns a false validation result if terminal spaces are not connected because of the lack of doors or openings (C). If only one route with unidentifiable validation satisfies the conditions, then CG-based checking returns an unidentifiable validation result-(D).

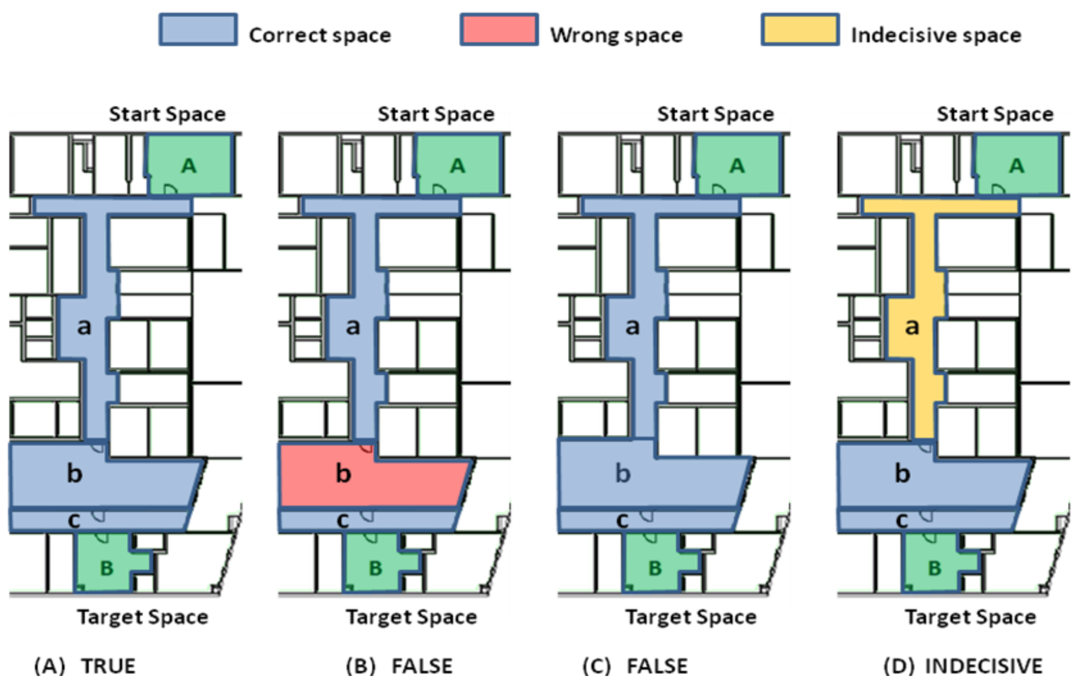


Figure 44 Four cases of validation for connectivity graph

AG-based checking also has three types of validations—true, false, or unidentifiable—based on the validation of spaces in a route in the AG. It returns the same validation results as the CG. The only difference is that it is based on the route in the adjacency graph. Thus, case C in Figure 44 returns a true validation result in AG-based checking because spaces a and b are adjacent even if they are not connected by a door.

Final validation is done by integrating the two results of CG-based checking and AG-based checking. Combinations of the three cases (true, false and unidentifiable) in CG-based checking and AG-based checking produce nine possible cases. Final validation is true in both CG- and AG-based checking cases because at least one route satisfies all the required conditions (1).

Table 15 Final validation table with the connection and adjacency graphs

Case	Connection graph based checking	Adjacency graph based checking	Final validation comments
1	True	True	True
2	True	Potential	Checking error
3	True	Unidentifiable	Checking error
4	True	False	Checking error
5	Potential true(false)	True	Currently there is a route through potentially valid space, but by adding more doors, current design can have a valid route.
6	Potential true(false)	Potential	Currently there is a route through potentially valid space.
7	Potential true(false)	Unidentifiable	Checking error
8	Potential true(false)	False	Checking error
9	Unidentifiable	True	Currently there is a route through an unidentifiable space, but by adding more doors, it can have a valid route.
10	Unidentifiable	Potential	Currently there is a route through an unidentifiable space, but by adding more doors, it can have a route through a potentially valid space.
11	Unidentifiable	Unidentifiable	Currently there is a route through an unidentifiable space.
12	Unidentifiable	False	Checking error
13	False	True	Currently there is no valid route, but by adding more doors, it can have a valid route.
14	False	Potential	Currently there is no valid route, but by adding more doors, it can have a route through a potentially valid space.
15	False	Unidentifiable	Currently there is no valid route, but by adding more doors, it can have a route through an unidentified space.
16	False	False	Currently there is no valid route, and a current design does not have a potential to have a valid route.

Final validation is false in CG-based checking and true in AG-based checking because no correct route satisfied in the conditions in the current model (2). However, case (4), whose validation is false in both CG- and AG-based checking because case (2) has a potential to be true. Case 3, 5, 6, and 8 produce checking errors because they cannot occur if the checking is done correctly. The validation results from AG-based checking cannot be worse than those from CG-based checking because AG contains all the same routes as CG; thus, AG will also check the good routes in CG.

Cases 7 and 9 are both unidentifiable because they are unidentifiable in CG-based checking. Even though case 7 has a correct route in AG, and case 9 has still an unidentifiable route in AG, it is difficult to determine which case has more chance of becoming true because such a chance depends on the way an unidentifiable space is developed.

- **The start and target space condition**

In the checking of terminal space conditions, a terminal space could be either an aggregated space or a general space. For instance, a start space could be a courtroom in a model instead of a district or bankruptcy courtroom. From the set perspective, an individual element is an element of an aggregated or general space. Thus, in case the checking of an aggregated or general space returns false, then the checking of all the individual spaces under them will also return false. Thus, this thesis uses an aggregated or general space in a building model if there is no individual space. Then, the checking result is applied to all individual spaces. if the checking with aggregated or general space returns false, then the checking of all individual spaces under these spaces are false. In case the checking is true, then, the checking of all individual spaces has potential to be true.

6.3 The final algorithm for checking circulation rules

The final algorithm of rule with CG- and AG-based checking is described by using pseudocode.

```
IF
  Startspace = FindStartSpace();
  TargetSpace = FindTargetSpace();
THEN
   resultbyCG = CheckwithCG(Startspace, TargetSpace, TransitionCondition)
   resultbyAG = CheckwithAG(Startspace, TargetSpace, TransitionCondition)
  FinalResult = MakeaFinalDecision(resultbyCG, resultbyAG);

  Return FinalResult
ELSE
   Return Error.
```

6.4 Summary

This chapter explains how courthouse design in the development stage can be checked by using the generic method described in Chapter 5. This chapter describes how each circulation condition can be validated in each possible case of a courthouse model in development. The cases involve several types of spaces such as aggregated spaces, general spaces, and fully- developed spaces. The cases also consider potential space connections in the development of a design. Final validation is done through a review of the results of all of these possible cases.

CHAPTER 7

IMPLEMENTATION

The theory for the automated checking method is applied to the real validation of courthouse designs against occupant circulation rules in the *U.S. Courts Design Guide*. As a basis of implementation, object-oriented class structures, which are based on abstract logical representations of circulation rules, are implemented on top of the implementation platform, and space connectivity and adjacency information in a courthouse model are implemented by using the connectivity and adjacency graphs.

7.1 An object-oriented class structure for circulation rule checking

The implementation of the logical process of a generic method necessitates a definition of an object-oriented class structure. The class structure is developed to represent diverse circulation conditions and their combination with logical connections. As described above, a circulation rule consists of diverse conditions such as the security zone condition. The security condition is one that is assigned to one type of building element–space. The condition associated with a building element is referred to as an ***object condition***. An object condition is related to other object conditions with logical connections such as AND or OR. The condition that defines the logical relationship among object conditions is referred to as a ***logical condition.*** Another type of condition is the ***cardinality condition,*** which defines quantification requirements in a rule. Thus, conditions can be classified three types: an ***object condition***, a ***cardinality condition***, and a ***logical condition*** (see Figure 45).

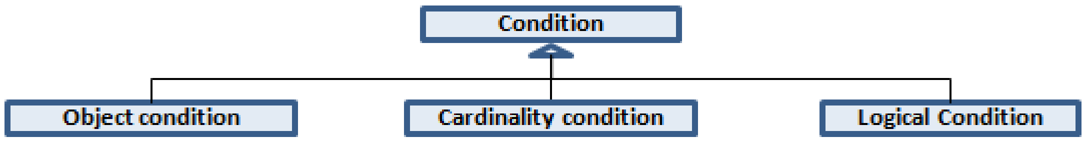


Figure 45 The class structure of the conditions

An object condition is one associated with a building element in a rule. Most circulation rules pertain to building elements related to circulation such as a space, a door, or an elevator. An object condition also can be classified into two types: a **single-object condition** and a **composite-object condition** (see Figures 46 and 47)**.** A single-object condition is one that is assigned to a building element such as a security zone condition on a space. However, some conditions are related to a combination of building elements. For instance, the distance condition can be assigned to a route, which is a combination of spaces on the route. The condition related to a combination of building elements is referred to as a composite-object condition.

A single-object condition has many sub-types according to the type of building element. Two sub-types of the single-object condition for circulation rule checking are the space-object condition and the door-object condition. The sub-types can be developed by extending the coverage of a circulation rule. Currently, for a circulation rule checking, a space-object condition consists of the following conditions: a space-name condition, a space-zone condition, a space-usage condition. For example, a space-object condition also can be extended according to the type of space properties for the purpose of dealing with a space-area condition and a space-height condition (see Figure 46).

According to the *U.S. Courts Design Guide,* the circulation rule for checking a door object is that it meets the door opening direction condition. The condition for a door also can be extended to other conditions such as the door width condition or the door height condition for checking the circulation rules pertaining to accessibility through a door.

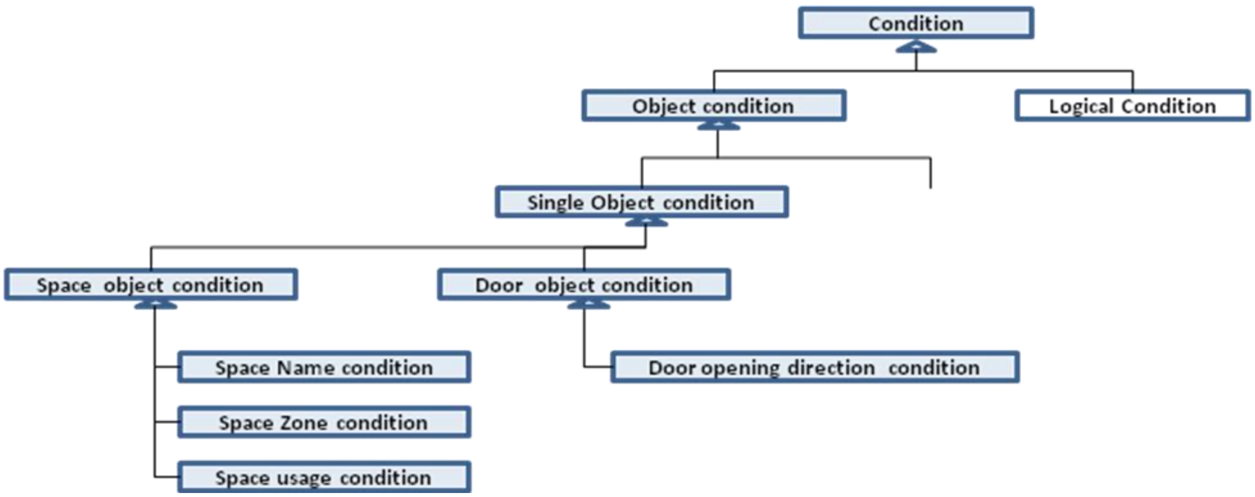


Figure 46 The class structure of an object condition

A composite-object condition includes conditions on the composition of building elements such as a route (see Figure 47). A route is an important concept for circulation rule checking because circulation rule checking specifically requires identifying a valid route. However, a composite- object concept is also useful for other type of rules associated with the grouping of building elements such as checking the areas of a department, or visibility checking through spaces. For circulation checking, the composite-object condition has a ***route-object condition***. A route-object condition, which is an abstract class for diverse conditions on a route, is divided into the following sub-classes: a route-length condition, a direct-access condition, a vertical-access condition, and a required-space condition.

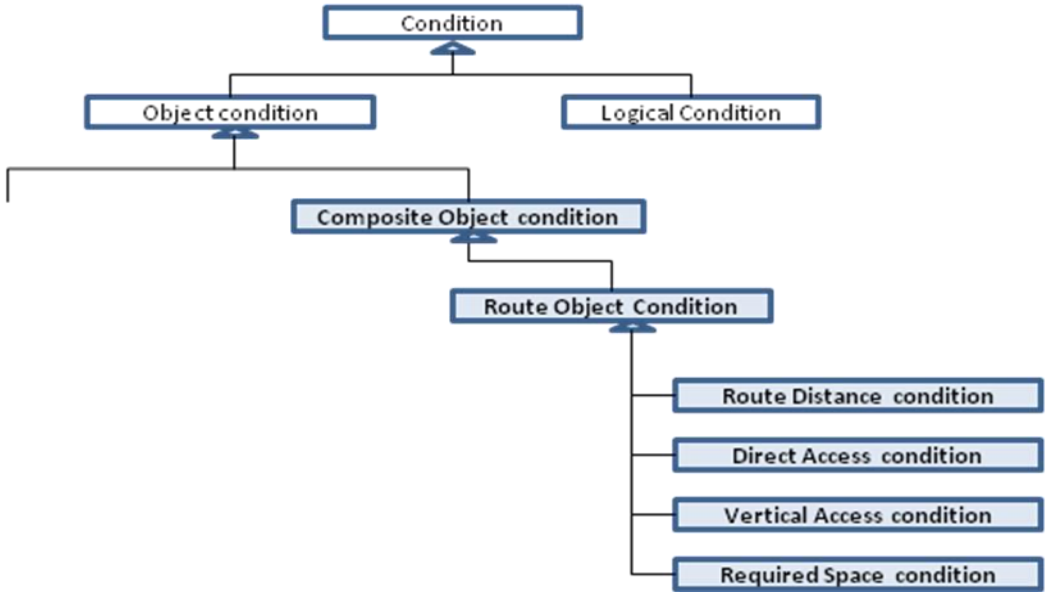


Figure 47 The class structure of a composite object condition

The logical condition is classified into two main types: a ***definitive logical condition*** and a ***connective logical condition***. A definitive logical condition is a type of logical condition that defines one sentence such as NOT, AT LEAST ONE, and ALL.

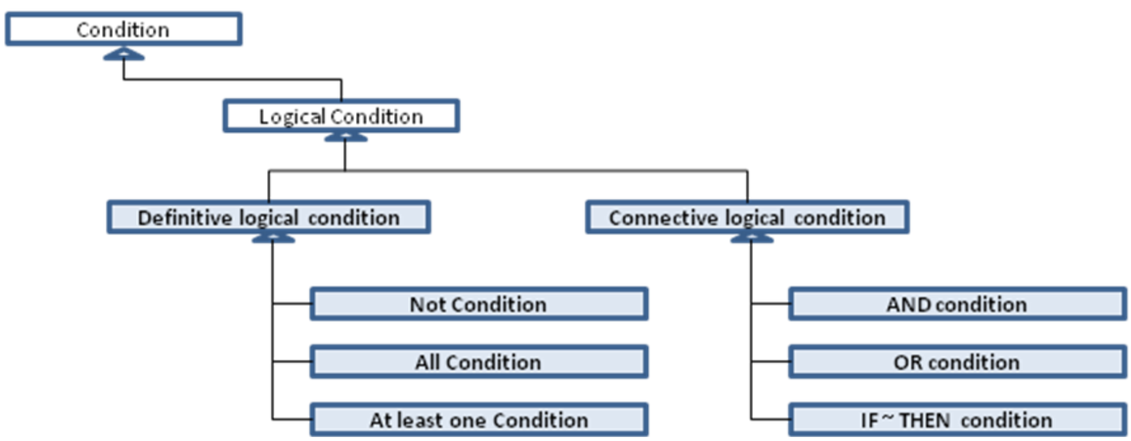


Figure 48 The class structure of a logical condition

By contrast, a connective logical condition defines two or more sentences with an AND or an OR relationship (see Figure 48).

- **The relationship between classes**

A rule has a relationship of one to one or one to many in a condition because a rule consists of many conditions. Since a definitive logical condition defines one object condition, and a connective logical condition defines the logical relation of two or more object conditions, a definitive logical condition has a one-to-one relationship to an object condition, and a connectivity logical condition has one-to-many relationship to an object condition (see Figure 49).

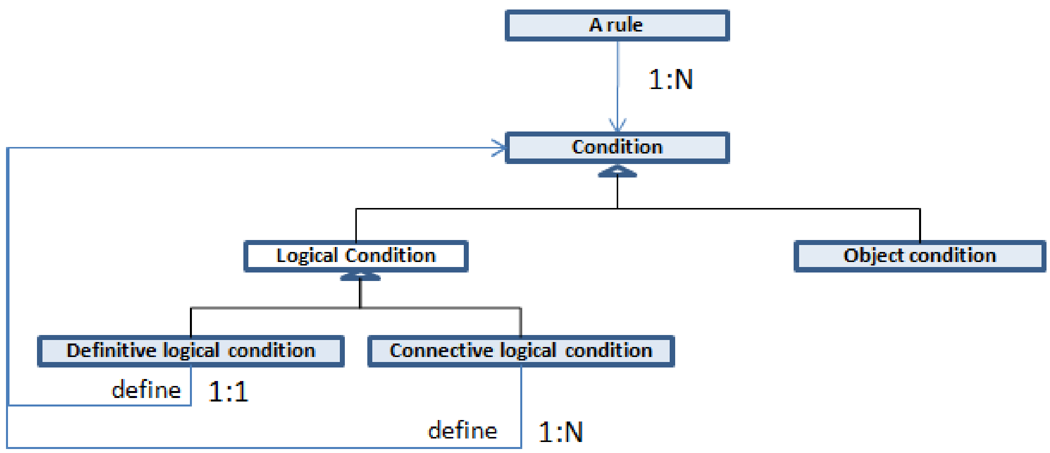


Figure 49 The relationship between a logical condition and a condition

However, this class structure does not support the concept of grouping conditions. Thus, this class structure is developed using the concept of ***atomic condition*** and ***composite condition***, the latter of which is the composition of atomic conditions (see Figure 50).

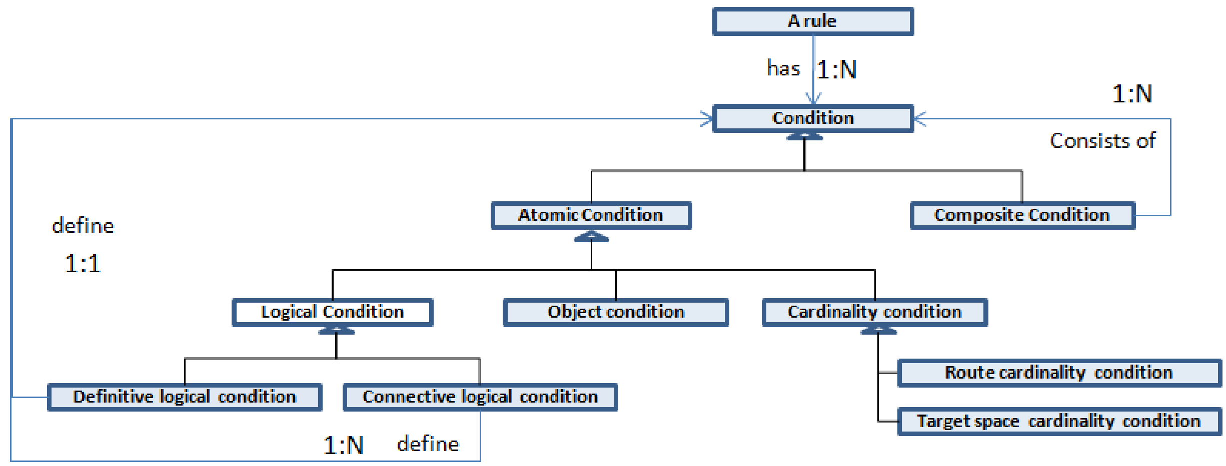


Figure 50 The relationship between an atomic condition and a composite condition

A cardinality condition defines the quantity requirements for validation. It could be any mathematical quantification such as one, more than one, less than ten, and so forth. Thus, a cardinality condition contains descriptions of the quantification requirements for validation. For circulation rule checking, a cardinality condition can be one of two types: a route cardinality condition for the quantification of routes, or a target space cardinality condition for the quantification of a target space (see Figure 50).

A composition condition has two sub-classes: a terminal condition for the start and target space condition in parameters, and a transition condition for the combination of the conditions in traversing (see Figure 50).

7.2 The System Structure of the Generic Checking Module

A generic checking module consists of mainly three parts: a graph generation module, a rule checking module, and a final validation module. The graph generation module, which works to build a graph from a courthouse model written using IFC, consists of two parts: an adjacency graph generation module and a connectivity graph generation module. The rule checking module, which checks circulation rules by using graphs generated by the graph generation module, is also composed of two parts: a rule checking module on an adjacency graph and a rule checking module on a connection graph. Each module traverses a graph to find a valid route, satisfying the required conditions in the rules. The last module is the final evaluation module, which makes a final evaluation by using the results of checking on the adjacency graph and those of checking on the connectivity graph.

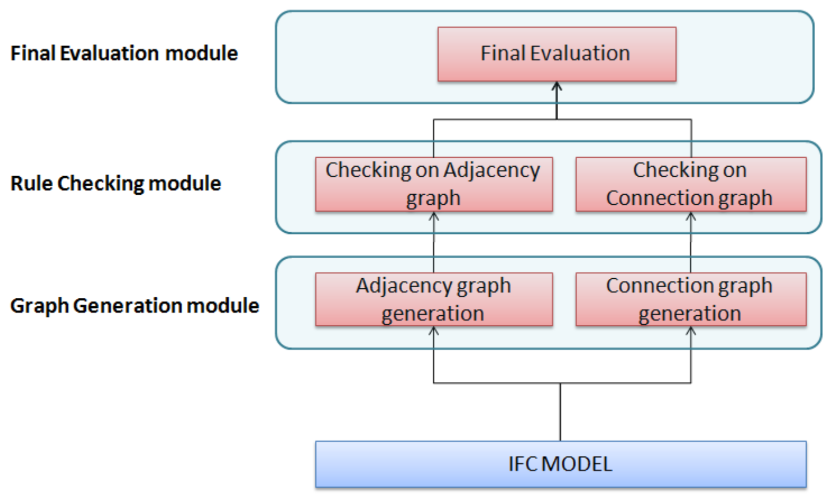


Figure 51 The structure of modules in the generic method

7.2.1 The Graph Generation Module

The main role of the graph generation module is to generate a graph for rule checking. This module can produce two types of graphs: an adjacency graph and a connectivity graph. The generation of a connectivity graph, which represents connections of spaces through elements such as doors or openings, was described in Chapter 5. Thus, this section will focus on the generation of an adjacency graph from an IFC building model.

An IFC building model contains explicit space connection information. By using the information, a connectivity graph can be generated. However, an IFC building model does not involve explicit space adjacency information. Thus, adjacency relationships are derived by using wall sharing information in the IFC because all adjacent spaces have at least one shared wall. The length of a shared wall is also considered to determine the adjacency of two spaces because walls of too short a length could not accommodate a door.

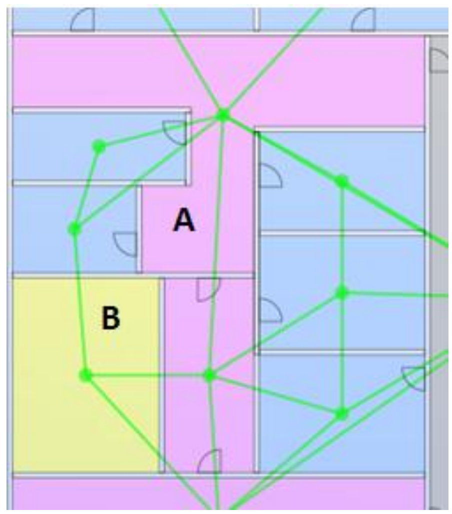


Figure 52 An adjacency graph

Figure 52 shows an adjacency graph generated by knowledge of the length of a shared wall. Spaces A and B do not have an adjacency connection because the length of the shared wall is too small. Currently, the minimum length for space adjacency is 80 cm because the minimum width of a door is 75cm.

This adjacency graph generation module is tested for generating an adjacency graph for a real courthouse model, the courthouse 1-model. It generates an adjacency graph without any problems. Note that the adjacency graph represents a direct space-to-space connection while the connectivity graph represents a space-to-space connection through an element such as a door or an opening.

7.2.2 Rule checking module

A rule-checking module is one that traverses a graph to validate the spatial relationship against required conditions in the rules. A theoretical checking process is used to check the existence of a valid route among all the routes between start and target spaces. However, in real implementation of the checking module, it does not identify all of the routes between two spaces that would be valid routes. Finding all routes is a very costly process because the number of routes increases exponentially as the number of nodes in a graph increases. Instead of finding all the routes, Dijkstra's algorithm [45] on the weighted graph is used to find valid routes. Dijkstra's algorithm traverses a graph by a depth-first search from the start space node until it locates a target space node. If it finds a space node that does not satisfy a required condition, it assigns a big weight value to the node. After that, it repeats searching until it finds the cheapest routes between the start and target spaces. However, each rule has different requirements, so the weighting of a space node also depends on the requirements of the rules.

Therefore, according to the rule requirements, the weighting of the space nodes should be done dynamically, and based on this dynamically assigned space weight, the search for the cheapest route is performed. In the implementation of the traversing of the weighted graph , the "closest first iterator" is chosen. Basically, this first iterator is a type of breadth first iterator. Thus, if a graph is not weighted, it iterates the adjacency nodes at the same depth first until it visits all of the nodes. When iterator visits a node, it assigns weight to a node according to the conformation of required conditions. After weighting all nodes, the closest first iterator traverses to a cheapest route based on the accumulated weights in each route until it reaches the target node.

As an example, if spaces 1,3,4, and 5 are assumed to be in the red zone and space 2 in the blue zone (see Figure 53), they will be connected in a similar way to those in the building model in Figure 53. If the rule requires accessibility through a red zone only, then the closest first iterator starting from space node 1 selects space node 3 because space node 2 has more weight than space node 3. This selection process is repeated until it reaches the target space—space node 4.

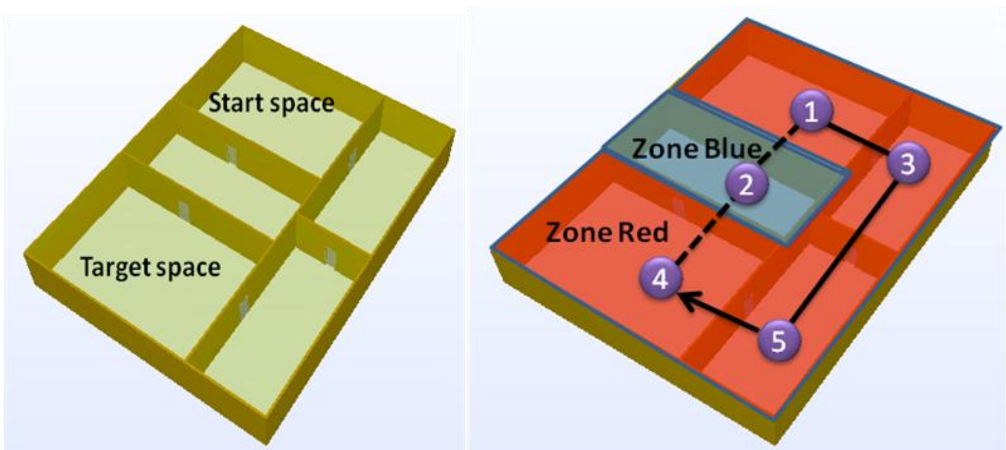


Figure 53 Traversing a graph by the closest first iterator

However, weighting a space node totally depends on the rule requirements. In this case, a high weight value is assigned to space node 2 because the rule requires accessibility through a red zone only. So that the rule requirements are reflected in the weighting process of a graph, the weights are assigned to the adjacent node before the selection of the next traversing space node. For instance, before the selection of space node 3 from space node 1, the adjacency space nodes from space node 1(in this case, space nodes 2 and 3) are visited and weighted according to the rule requirements, and then the next node is selected as the cheapest weighted node.

If a building model does not have a route that satisfies the required condition in the rules between the start and target spaces, the checking system returns the cheapest violation route as a traverse route. The start and target spaces in the building model in Figure 54 are isolated by a blue zone. Thus, the model does not have a route through a red zone only. If the rule requires a route to be through a red zone only, this space layout cannot satisfy the required condition.

As space node 3 is in the blue zone, route traversing first entails going to space node 2 from space node 1 instead of going to space node 3.In the next step, the closest-first iterator compares the weight of traversing through space nodes 1,2, and 4 and the weight through 1 and 3. Then it selects routes 1 and 3 because they have less weight than routes 1,2, and 4. Finally, the system traverses a route through space nodes 1,3, and 5 as a cheapest violation route.   The image on the right side of Figure 54 shows the cheapest route returned by the automated circulation checking system.

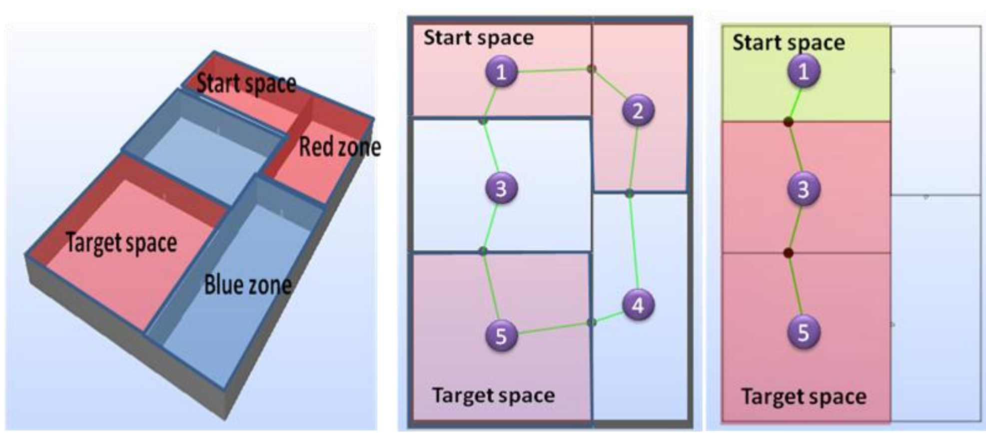


Figure 54 Traversing a graph through zones

7.2.3 The final validation module

The final validation module makes a decision by using the checking results from AG- and CG-based checking. It generates a final assessment on checking based on the final validation table (Table 13) in Chapter 6.

7.3 Summary

To implement the generic method, an object-oriented class structure for the generic method is defined. The classes are used to support diverse conditions such as space conditions, cardinality conditions, transition conditions, and their logical connections. These classes are used to implement main modules in the generic method. The generic method consists of three modules: the graph generation module, the rule checking module, and the final validation module. The graph generation module produces both connectivity and adjacency graphs automatically from a courthouse model written in IFC. Then the rule checking module performs circulation rule checking with these two types of graphs, and the final validation module generates final comments with the results of the checking with the two graphs.

CHAPTER 8

VALIDATION

8.1 Overview

Based on the theoretical validation of the rule checking process, this thesis does validation on the implementation of an integrated method to identify its correspondence with the purpose of thesis: to develop an automated circulation rule checking method that is specifically capable of checking an incomplete model during the developmental stage. Validation of this thesis also focuses on assessing and validating the planned capabilities. First, in order to validate the capability of checking an incomplete model in the development stage, the checking module is tested in two ways: on a test model and on real courthouse models. The test model produces ten theoretically possible models in the development stage that are used to show the capability of an integrated method of checking. Finally, real courthouse designs in the development state are tested with the generic method.

8.2 Validation of the generic method with the test model

A test model with a combination of fully developed spaces and spaces still in development was created. The test model includes fully developed spaces such as a start space, a target space, several offices, and a corridor. It also has spaces in development such as an aggregation space-support area, which will be developed into several subspaces but no named space (marked as “??”). In the test model, space connections through doors are also still in development. Since a space-support area does not have a door to a corridor, a door is not in the test model. However, more doors can be added to a support area or anywhere else in the design (see Figure 55).

This thesis adopts this model to validate the generic method. For testing purposes, this thesis tests a simple transition condition—the space usage condition (circulation space condition) between the start and target spaces—and compares the results from the three validation methods of graph-based checking, set-based checking, and generalized method-based checking.

Rule: Accessibility between a start and a target space through circulation spaces.

In graph-based checking, only the route through an existing connection is checked and validates spaces in binary terms: true or false. Thus, spaces in the design are classified into only two types: circulation space and non-circulation space. The following is the classification for graph-based checking of the circulation condition among the spaces in figure 55.

Circulation space = { restricted corridor }

Non-circulation space = {all other spaces than circulation spaces}

Based on the classification, spaces in the test model can be classified into spaces shown in Figure 55. Checking of the rule on the previous page returns a false because two terminal spaces are not connected through circulation spaces (restricted corridors in Figure 55).

Set-based checking assumes that all spaces are accessible if they have an adjacency relationship through spaces (note that the set-based method creates space-sets by using a security level only). In the test design, all the spaces are accessible because they have adjacency relationship through spaces adjacent between them. Thus, the validation of the circulation condition is true.

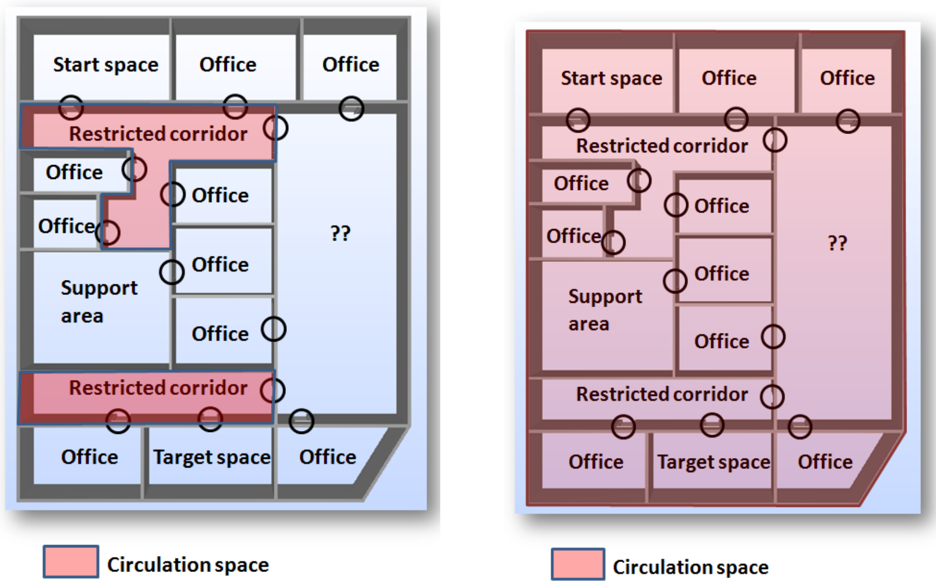


Figure 55 Accessible spaces in the connection and adjacency graphs

However, the generic method classifies spaces to satisfy the circulation condition into four types: a circulation space (a true case), a non-circulation space (a false case), a potential circulation space (a potential case), and an unidentifiable space (an unidentifiable case). For the test design, spaces are classified into the following (see Figure 58).

Circulation space ={ restricted corridor }

Non-circulation space = {office, start space, target space}

Potential circulation space = {support area}

Unidentifiable space = {??}

The generic method checks the circulation condition through two graphs: a connectivity graph and an adjacency graph. Connection graph-based checking returns a route through a start space, a restricted corridor, an unnamed space ("??"), a restricted corridor, and a target space. The validation of this route is "unidentifiable" because it includes an unidentifiable space. Adjacency graph-based checking returns a route through a start space, a restricted corridor, a support area, a restricted corridor, and a target space because it is a minimum weighted route in the adjacency graph. Validation of this route is "potential" because it has a space classified as potential (the support area). Thus, the final validation comment about this design is

> "***Two spaces can be connected potentially by adding doors to the support area, but it is currently connected through an unidentified space.***"

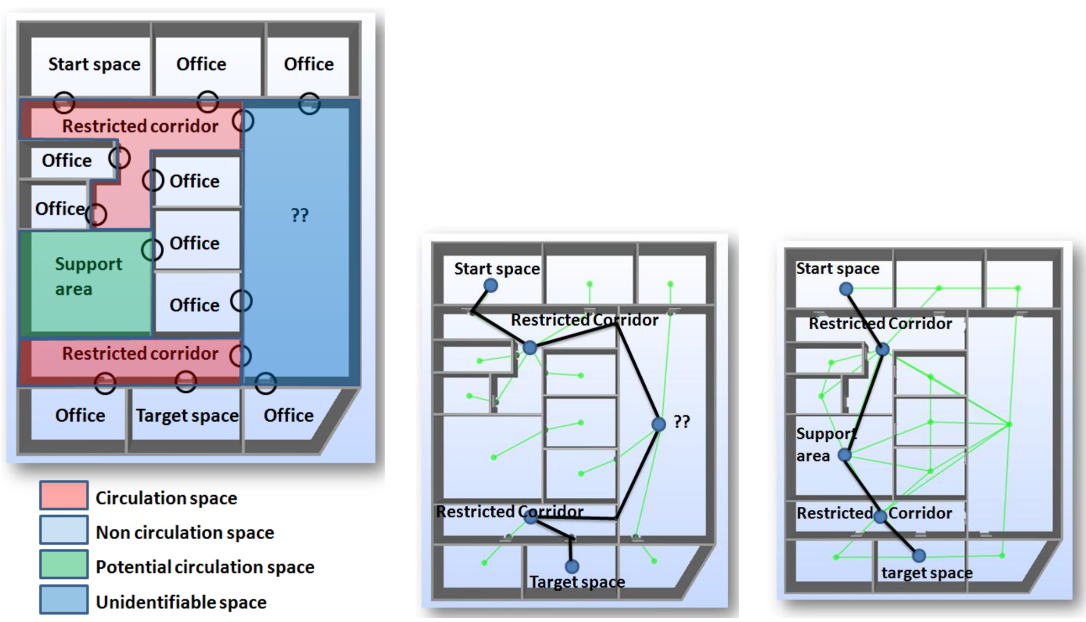
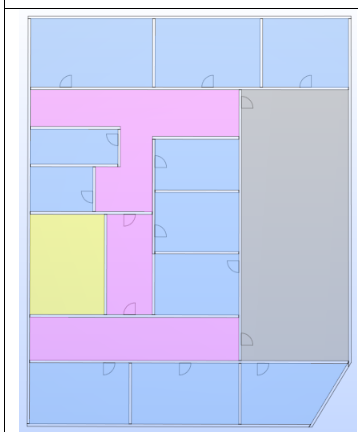
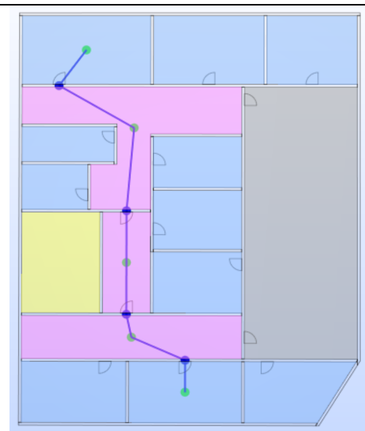
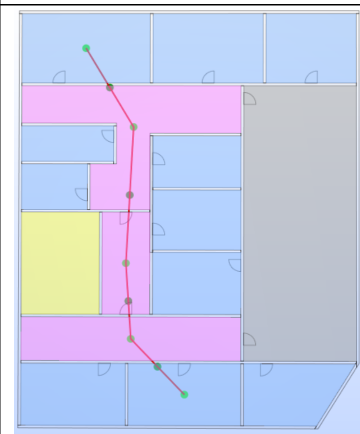


Figure 56 Connection and adjacency graphs in the generic method

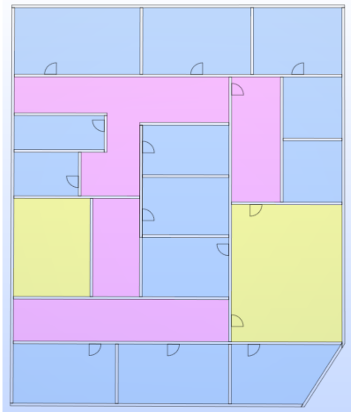
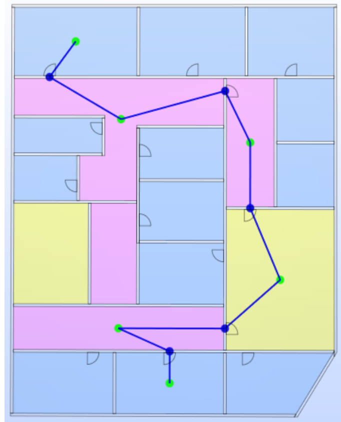
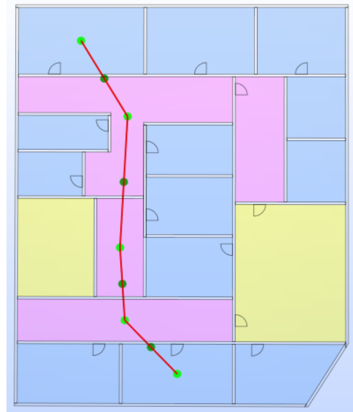
The test model is then checked with the generic checking method, which evaluates the model successfully as expected. As described in Chapter 6, validation through the generic method could produce sixteen cases because connection graph-based checking can produce four cases (true, potential, unidentifiable and false) and adjacency graph-based checking also four cases. Among the sixteen cases, only ten are valid because the adjacency graph produces fewer validation results that satisfy the condition than the connectivity graph.

Models for the ten valid case models are created and tested. The spaces in the test models have four colors, each of which represents one of the validation results against the space usage condition (i.e., the circulation condition). The blue space is the false-validation space such as office and a toilet against the circulation condition. The red space is a true-validation space such as a corridor. The gray space is an unidentifiable-validation space such as an unnamed space. The yellow space is a potential-validation space such as a building support area, which can be both a circulation space and a non-circulation space according to the design development.

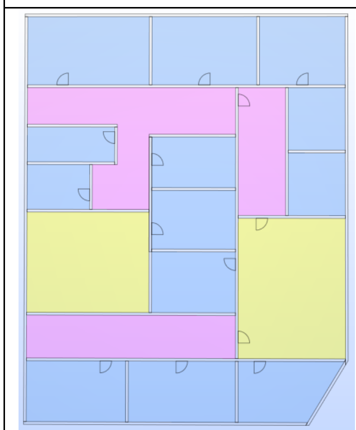
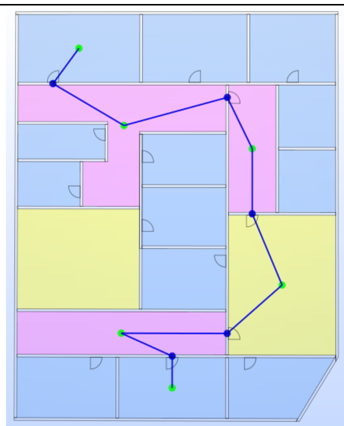
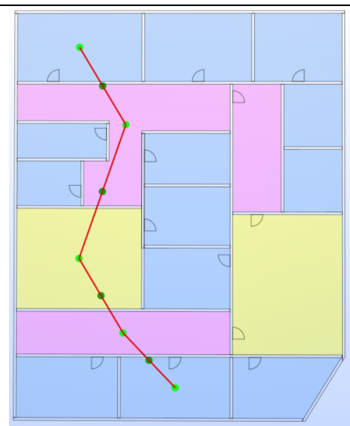
Case 1: CG—True and AG—True [CG: connection graph, AG: Adjacency graph]

Space Classification	CG-based checking	AG-based checking
		
Validation result	True	True
Final comment	Description There is a path that uses following transition conditions starting from START SPACE(start space [1] of Floor Level 1) and ending TARGET SPACE(target space [14] of Floor Level 1). Transition conditions are Usage: circulation.	

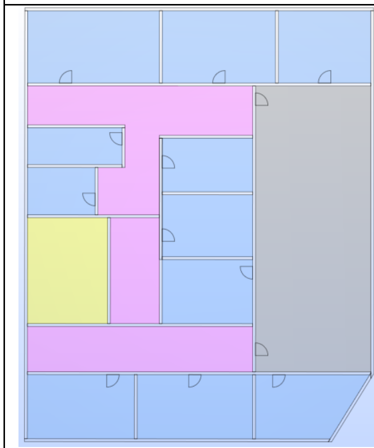
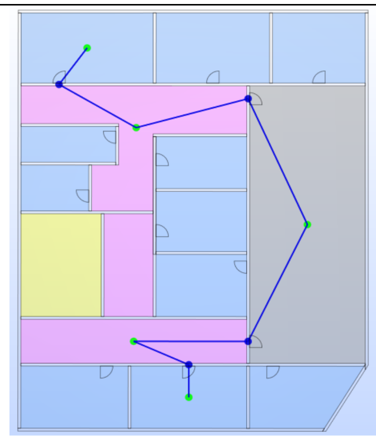
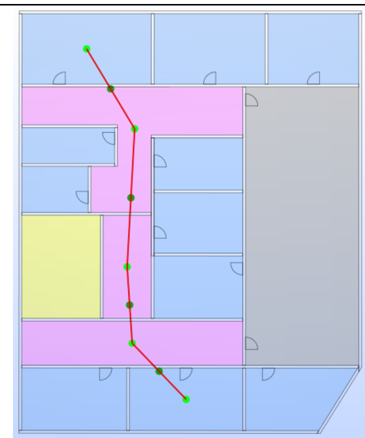
Case 2: CG—Potential true and AG—True

Space classification	CG-based checking	AG-based checking
		
Validation result	Potential True	True
Final comment	Description Currently there is a route between START SPACE(start space [1] of Floor Level 1) and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1) through potentially valid space, but by adding more doors, current design can have a valid route.	

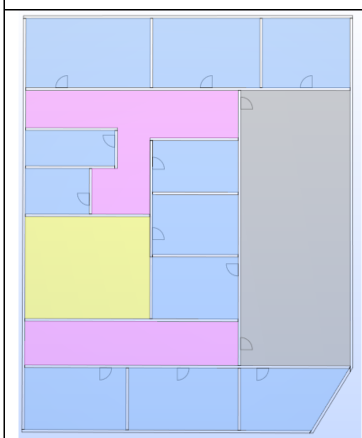
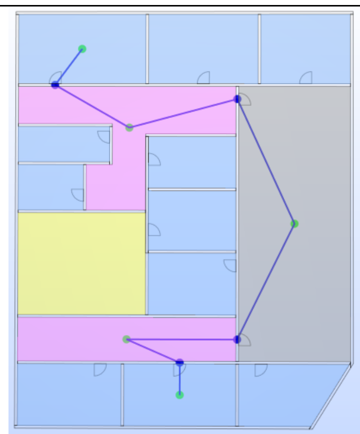
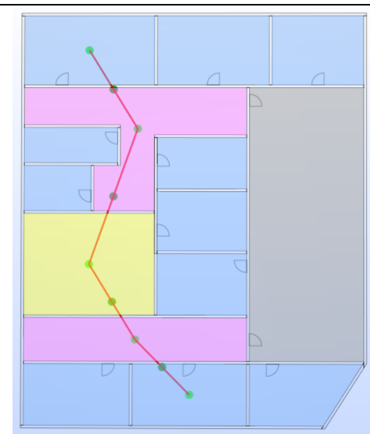
Case 3: CG—Potential true and AG—Potential true

Space classification	CG-based checking	AG-based checking
		
Validation result	Potential true	Potential true
Final comment	Description Currently there is a route between START SPACE(start space [1] of Floor Level 1) and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1) through potentially valid space.	

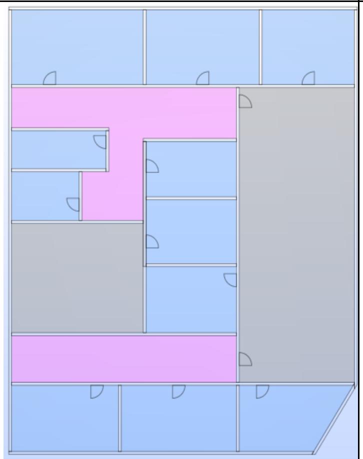
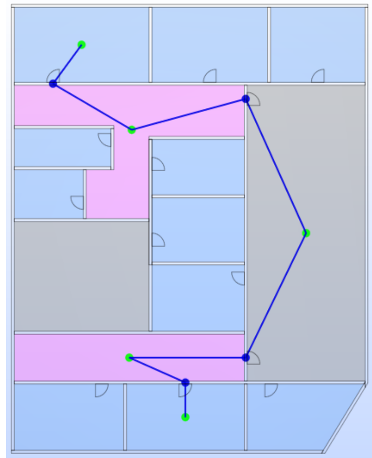
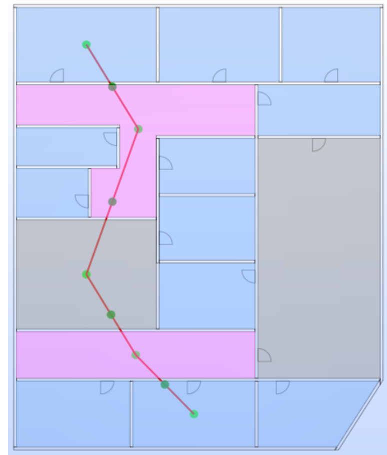
Case 4: CG—Unidentifiable and AG—True

Space classification	CG-based checking	AG-based checking
		
Validation result	Unidentifiable	True
Final comment	Description Currently there is a route between START SPACE(start space [1] of Floor Level 1 and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1) through an unidentifiable space, but by adding more doors, it can have a valid route.	

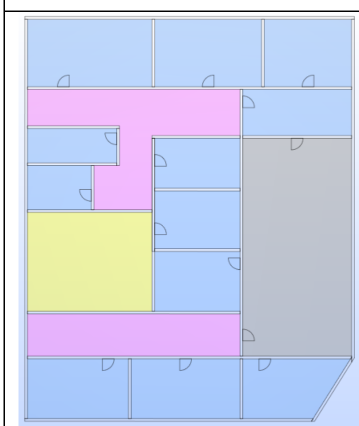
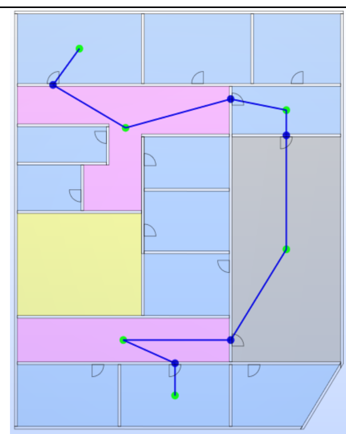
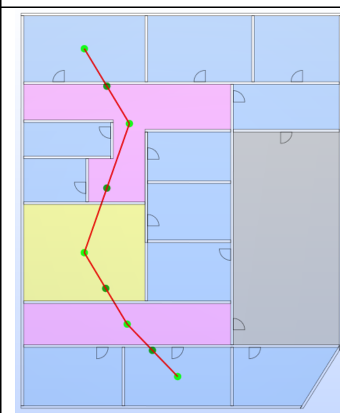
Case 5: CG—Unidentifiable and AG—Potential true

Space classification	CG-based checking	AG-based checking
		
Validation result	**Unidentifiable**	**Potential true**
Final comment	Description Currently there is a route between START SPACE(start space [1] of Floor Level 1) and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1) through an unidentifiable space, but by adding more doors, it can have a route through a potentially valid space.	

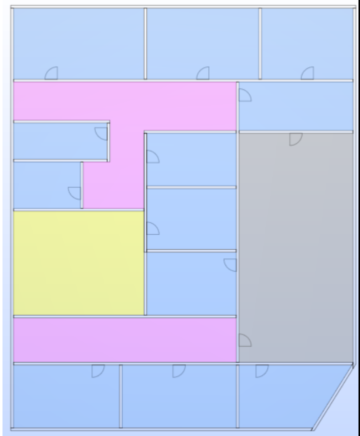
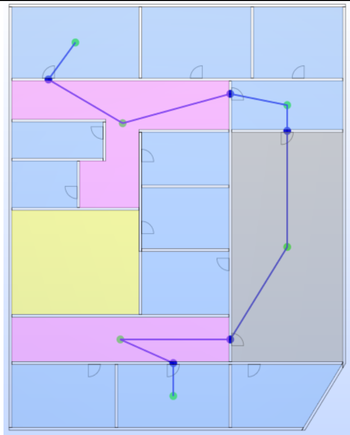
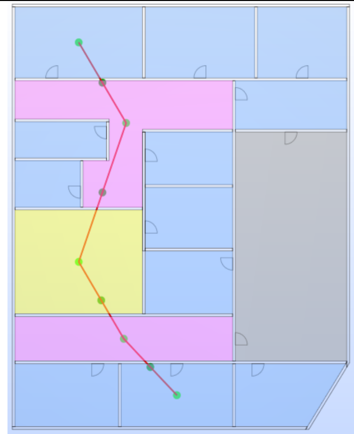
Case 6: CG—Unidentifiable and AG—Unidentifiable

Space classification	CG-based checking	AG-based checking
		
Validation result	**Unidentifiable**	**Unidentifiable**
Final comment	Description Currently there is a route between START SPACE(start space [1] of Floor Level 1 and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1) through an unidentifiable space.	

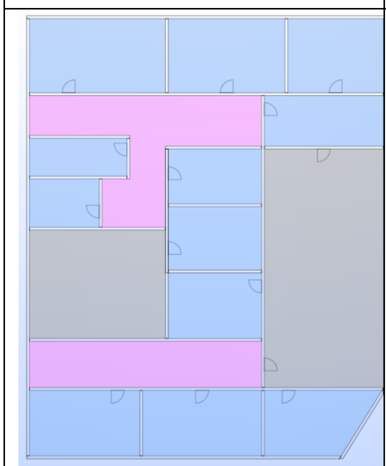
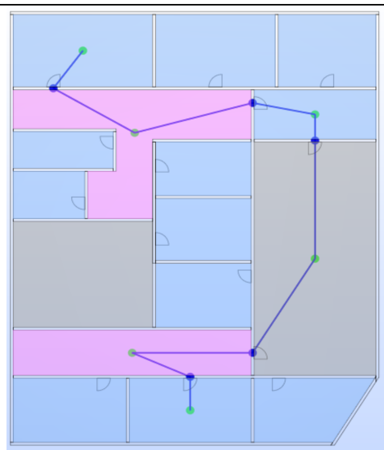
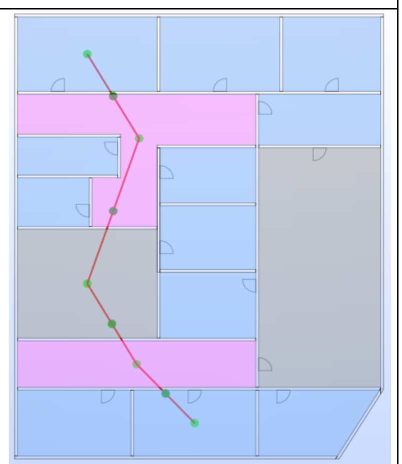
Case 7: CG—False and AG—True

Space classification	CG-based checking	AG-based checking
		
Validation result	**False**	**True**
Final comment	Description Currently there is no valid route between START SPACE(start space [1] of Floor Level 1) and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1), but by adding more doors, it can have a route through a potentially valid space.	

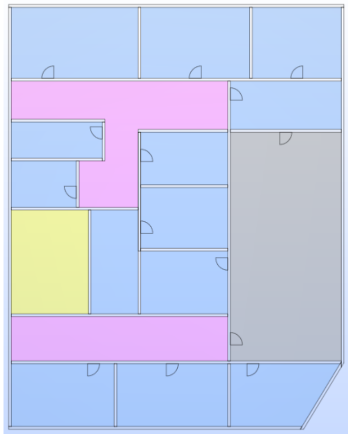
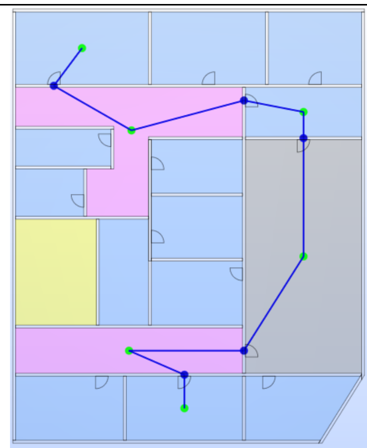
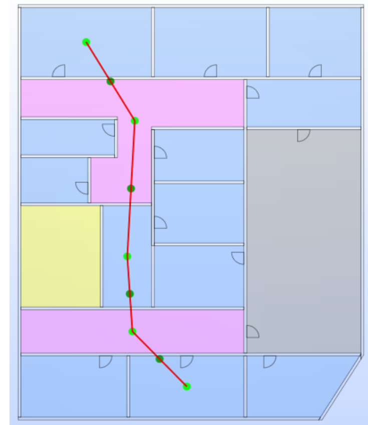
Case 8: CG—False and AG—Potential true

Space classification	CG-based checking	AG-based checking
		
Validation result	**False**	**Potential true**
Final comment	Description Currently there is no valid route between START SPACE(start space [1] of Floor Level 1) and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1), but by adding more doors, it can have a route through a potentially valid space.	

Case 9 : CG—False and AG—Unidentifiable

Space classification	CG-based checking	AG-based checking
		
Validation result	**False**	**Unidentifiable**
Final comment	Description Currently there is no valid route between START SPACE(start space [1] of Floor Level 1) and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1), but by adding more doors, it can have a route through an unidentified space.	

Case 10 : CG—False and AG—False

Space classification	CG-based checking	AG-based checking
		
Validation result	**False**	**False**
Final comment	Description Currently there is no valid route, and a current design does not have a potential to have a valid route between START SPACE(start space [1] of Floor Level 1) and TARGET SPACE(TARGET SPACE [14] OF FLOOR LEVEL 1).	

As shown above, the implemented generic method checks correctly on both the adjacency and connection graphs, and returns expected comments.

8.3 Validation of the generic method with real courthouse models

The generic checking method validates five courthouses designs: courthouse 1, 2,3,4 and 5. Some of the courthouses have multiple versions of their design during development. The courthouse 2 has two versions, and the courthouse 5 has five versions from preliminary concept design to final concept design. Thus, a total of ten courthouse models are tested by using the generic checking method.

Each courthouse model is in a different stage of completeness. Some models have fully-developed individual spaces, all internal wall, and doors; some models consist of undivided aggregated spaces, space boundaries without internal walls, and no doors. The models with individual spaces, internal walls, and doors are classified into final concept design models, and those with aggregated spaces, no internal walls, and no doors are classified as preliminary concept design models. The models that fall between a preliminary concept design and a final concept design are classified as a model in development.

Table 16 Classification of courthouses in the design stage

Courthouse Name	**Modelling Date**	**Modelling Status**			
		Space	Space Boundary	Door	Status
Courthouse 1	2009/08/09	Individual spaces	Internal wall	door	Final concept design
Courthouse 2	2007/07/17	Aggregated spaces + Individual space	Virtual wall	No door	Model in development
	2007/08/24	Individual spaces	Internal wall	Door	Final concept design
Courthouse 3	2009/11/12	Individual spaces	Internal wall	Door	Final concept design
Courthouse 4	2009/02/10	Individual spaces	Internal wall	Door	Final concept design
Courthouse 5	2008/05/27	Aggregated spaces + Individual spaces	Internal wall	No door	Model in development
	2009/05/12	Aggregated spaces + Individual spaces	Internal wall	No door	Model in development
	2009/07/24	Individual spaces	Internal wall	Door	Final concept design
	2009/08/07	Individual spaces	Internal wall	Door	Final concept design
	2009/12/10	Individual spaces	Internal wall	Door	Final concept design

Table 16 shows the classification of twelve courthouse models into preliminary concept design, final concept design, and model in development.

Three types of checking methods—a topological graph-based checking method, a set-based checking method, and a generic checking method—are applied for checking the models in the final concept design, preliminary concept design, and design in development stages. One version of the courthouse 2 model and two versions of the courthouse 5- models are tested with the set-based method; and the courthouse 1, 3 and 5 models in the final concept design stage are validated with the topological graph-based method.

Table 17 Application of the checking method to each courthouse model

Courthouse Name	Modelling Date	Modelling Status	Applied Methods for Checking		
Courthouse 1	2009/08/09	Final concept design	N/A	Graph-based method	Generic method
Courthouse 2	2007/07/17	Design in development	Set based method	N/A	Generic method
	2007/08/24	Final concept design	N/A	Graph-based method	Generic method
Courthouse 3	2009/11/12	Final concept design	N/A	Graph-based method	Generic method
Courthouse 4	2009/02/10	Final concept design	N/A	Graph-based method	Generic method
Courthouse 5	2008/05/27	Design in development	Set based method	N/A	Generic method
	2009/05/12	Design in development	Set based method	N/A	Generic method
	2009/07/24	Final concept design	N/A	Graph-based method	Generic method
	2009/08/07	Final concept design	N/A	Graph-based method	Generic method
	2009/12/10	Final concept design	N/A	Graph-based method	Generic method

The generic method checks all types of models in the final concept design, preliminary concept design, and design in development stages. Thus, it is applied to all ten versions of the courthouse models. Table 17 shows the application of the checking methods according to the type of model. For instance, the courthouse 1 is a model of the final concept design stage, so the (topological) graph-based method and generic methods are applied to the checking process.

Three checking modules return checking results with graphics and comments on the individual results. The graphics of the results describe each checking-result with a route that has at least one space violating the required conditions (see Figure 57). The route in the figure is a less weighted route among the routes violating the required conditions. "Less weighted" indicates that the route violates the required conditions less severely than the other wrong routes. The route is visualized as a continuing poly-line, and spaces with violations are red.

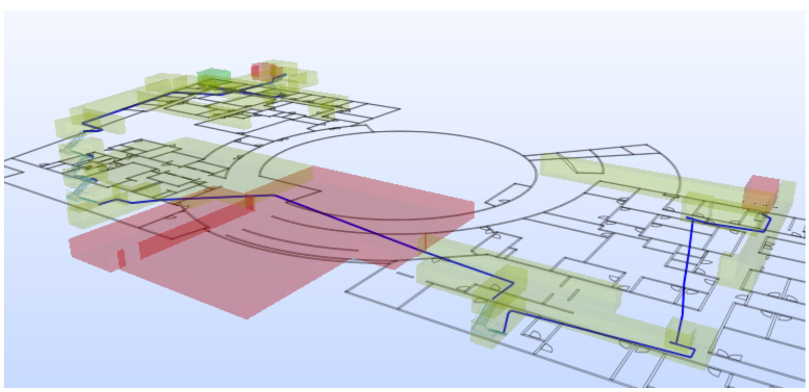


Figure 57 Visualization of an individual issue after checking

The comments about the results of checking state the existence of a valid route in the current connection, potential connectivity, transition conditions, evaluation results of the current traversing routes, and potential routes.

The following example shows comments on the connection between the bankruptcy clerk of court and the central court library after checking. The comments say that the two spaces do not have a route that satisfies the required transition conditions in either the current connection or the potential connection. It also says that the spaces do not satisfy the required conditions for either a current traversing route or a potential route.

Table 18 An example of comments about checking results

Currently there is no valid route, and a current design does not have a potential to have a valid route between BANKRUPTCY CLERK OF COURT(bankruptcy clerk of court office [2-111] of Floor Level 2) and CENTRAL COURT LIBRARIES(central library [3-881] of Floor Level 3). Transition conditions are Security Level: restricted,Usage: circulation,Route Length: 328'-1". Current route has : Components, which has a potential to be true for the "Usage: circulation" condition: Space.-4.6 : USMS Parking[P1-100] Components failing the : "Usage: circulation" condition: Space.-1.3 : Contracts Employees[3-150] Potential route has : Components failing the : "Security Level: restricted" condition: Space.-1.15 : Corridor[3-900] and Space.-1.20 : Lobby[3-001].

8.4 Validation of the Checking Results by the Generic Method

Each issue that was identified automatically is verified according to the following three approaches:

1) Comparing the results from three methods (i.e., the graph-based method, the set-based method, and the generic method)
2) Visualizing issues on each floor
3) Tracking traversing routes in validation.

1) Comparing the results from the three methods

First, the results are verified by comparing the results from the three checking methods. The number of issues identified by each method should satisfy the following criteria.

- *The number of issues identified by the set-based method should be smaller than that by the other methods.*

The set-based checking method checks only the security level conditions with an adjacency relationship. It does not consider other conditions in the circulation rules such as space usage conditions. Thus, the number of issues identified by the set-based checking method should be smaller than that by the other methods.

- *The number of issues identified by the graph-based method should be same as that by the generic method.*

The generic method validates a model by combining the validations results from both the connectivity and adjacency graphs. The number of issues identified by the adjacency graph is smaller than that by the connectivity graph because it searches valid routes with an adjacency relationship. All issues detected by the generic method should also be found by the connectivity graph. The adjacency graph validates the potential correction of issues that are detected from the connectivity graph. The graph-based method also uses a connectivity graph. Thus, the number of issues identified by the generic method should be same as that by the connectivity graph.

The checking result in Table 19 shows that the results of the eleven courthouse models satisfy these required criteria.

2) Visualization of issues per floor

The second level of validation from the checking results is done by visualizing the issues in each floor plan. Each issue detected by the generic method is visualized from three perspectives: the criticality of the issue, the type of issue, and the classification of spaces in required conditions. The perspective of the criticality of the issue shows problem spaces, which cause problem with the color coding that indicates such criticality. For instance, the space colored red in the first image in Figure 58 denotes a space that causes problems and has no potential to be corrected. The space colored yellow is a space that causes problems but has a potential to be corrected by a design changes during development. In the first image in Figure 58, a space (2-740) is a clerk's office that causes a circulation violation because it accesses a district clerk's office (a red circled space). A clerk's office is a private space that is typically not used for circulation, so this space is color-coded in red. A space (2-747) is a staff lounge, which also causes a violation of the circulation condition, but it is color-coded yellow because it has a potential to be used as a circulation space.

The second image in Figure 58 shows the issues by type of violation. Purple indicates a circulation condition violation, so we can see which space violates which condition.

The last image in Figure 58 visualizes the space based on the space usage condition. Circulation spaces are in pink, and private spaces are cyan. From this figure, we can see that the district clerk's office is not connected to any circulation space.   Instead, one has to pass through a private space—a clerk's office—to get there (2-740).

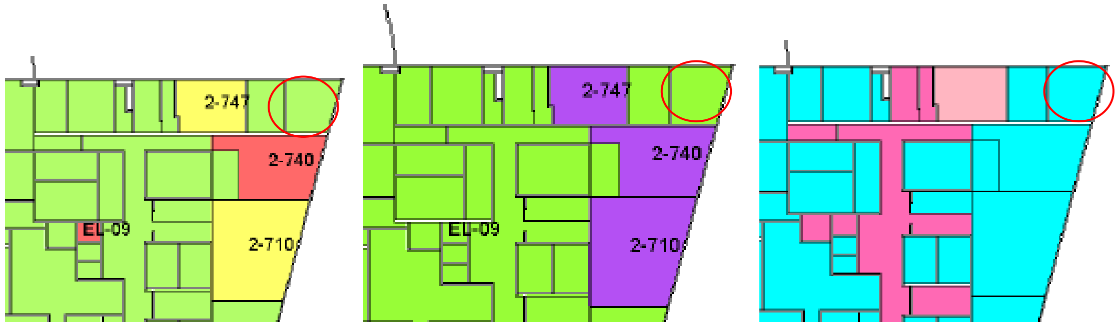


Figure 58 Verification of an issue with color-coded floor plans

The generic rule method can automatically generate a report that contains all the floor plans of a building with color-coded spaces from the three perspectives of the criticality of an issue, the type of issue, and the classification of spaces in the required conditions (see reference 48—a report example). By using the visualization of spaces in a floor with a different color code, many issues could be validated more intuitively.

3) Tracking traversing routes in validation

The most accurate way of validating each checking result is to track all the routes the system traversed when it checked a rule. Of course, such tracking is a huge job because the number of traversed routes in the checking process could considerable; thus, tracking should represent a final approach in validating a checking result. However, for more accurate validation, the development of a generic checking method should include the development of a reporting module that contains all the background information generated while rules are checked.   A module contains the name of a space in a courthouse model, an applied rule ID for the space, transition conditions in the applied rule, traversed routes, any conditions violated by the traverse route, a space list that causes violation, and a final comment based on the checking results. These data are used for a final validation of the checking results.

Reference 49 is attached to show an instance of background data that are generated while checking a sample courthouse model.

The validation of each courthouse against the circulation rules is summarized from the following three perspectives.

- ***The Number of issues (#issue)***

  In checking, a space could be checked by multiple circulation rules that could also be applied to multiple spaces. The number of checking instances varies according to the composition of the spaces in a courthouse. The number of issues is the total number of all of the individual problems automatically identified by the checking system.

- ***The number of spaces that cause circulation rule issues (Current connection / Potential connection) –(# pspace)***

  As described in the explanation of the comments, a wrong route has at least one space that does not satisfy the required conditions. "Then the number of spaces that cause circulation rule issues" is the total number of spaces that do not satisfy the required conditions. The number is calculated in the current connection (by using the connectivity graph) and the potential connection (by using the adjacency graph).

  In short, the space causing circulation rule issues is referred to as a "problem space."

- ***The number of rules that have issue(s)-(#rule)***

  This perspective is the number of total rules that has more than one violation.

The following table summarizes the checking results.

Table 19 The final checking results of the ten courthouses

Courthouse Name	Modelling Date	Modelling Status	Summary of Checking Results		
			Set-based method	**Graph-based method**	**Generic method**
Courthouse 1	2009/08/09	Final concept design	N/A	#issue: 112 #pspace: 33/1132 #rule : 21/183	#issue : 112 #pspace:33 (current)/1132 31 (potential) /1132 # rule : 21/183
Courthouse 2	2007/07/17	Design in development	#issue: 1 #rule : 1/183	N/A	#issue : 1 #pspace: N/A (current)/167 4 (potential) /167 # rule : 1/183
	2007/08/24	Final concept design	N/A	#issue : 48 #pspace: 10/420 # rule : 11/183	#issue : 48 #pspace: 10 (current)/420 10(potential) /420 # rule : 11/183
Courthouse 3	2009/11/12	Final concept design	N/A	#issue : 29 #pspace: 11/568 # rule : 9/183	#issue : 29 #pspace: 11 (current)/568 14(potential) /568 # rule : 9/183
Courthouse 4	2009/02/10	Final concept design		#issue : 43 #pspace: 12/159 # rule : 13/159	#issue : 43 #pspace: 12 (current)/159 10(potential) /159 # rule : 13/183
Courthouse 5	2008/05/27	Design in development	#issue: 21 #rule : 7/183	N/A	#issue : 31 #pspace: N/A(current)/221 18(potential) /221 # rule : 9/183
	2009/05/12	Design in development	#issue: 13 #rule : 7/183	N/A	#issue : 63 #pspace: 12(current)/357 15(potential) /357 # rule : 13/183
	2009/07/24	Final concept design	N/A	#issue : 136 #pspace: 27/1238 # rule : 10/183	#issue : 136 #pspace: 27(current)/1238 32(potential) /1238 # rule : 10/183
	2009/08/07	Final concept design	N/A	#issue : 160 #pspace: 33/1238 # rule : 10/183	#issue : 160 #pspace: 33(current)/1238 45(potential) /1238 # rule : 10/183
	2009/12/10	Final concept design	N/A	#issue : 173 #pspace: 44/1238 # rule : 10/183	#issue : 173 #pspace: 44(current)/1238 48(potential) /1238 # rule : 10/183

The generic checking method checks all eleven models and returns the checking results successfully. The courthouse 2 model (**2007/07/17**) returns only one error, partly because of its efficient space layout and partly because of its small number of spaces. However, the courthouse 6 model returns 296 issues among the 1,439 spaces.

8.5 Summary

The implemented generic method is tested in two different models: example models and real courthouse models. The generic method checks ten example models and eleven real courthouse models in a diverse design development stages, and the method found many issues in less than one minute.

The issues found by automated checking are validated in three ways: 1) Comparing the results from the three methods—the graph-based method, the set-based method, and the generic method; 2) visualizing the issues per floor; and 3) tracking all the individual traversing routes. The generic method was able to check all the courthouses in different stages during the design development process and found real design issues that were consistent with the results from other methods.

CHAPTER 9

DISCUSSION AND CONCLUSION

The generic method was developed to check a building model, including an incomplete building model over a range of development stages. This thesis develops a theoretical representation to describe and verify the checking process of the generic method. Then it tests and implements the theoretically described process, and demonstrates that it successfully checks the conditions against the rules in a range of design development stages. The implemented module works as expected, shown in the results of ten test cases and a real courthouse model. This chapter describes the findings of this thesis after the generic method was applied to determine the validation based on the rule checking in ten real courthouses and based on these findings, makes final conclusions about the results of this research.

9.1 Discussion issues from analyzing the checking results

After analyzing the checking results from the ten courthouses, this research found several issues worthy of discussion. One is the ratio of the number of issues identified by the connectivity graph to that by the adjacency graph. In Table 20, the number with parenthesis next to the number of issues (#issue) is the number of issues relating to adjacency connection. The number of issues in the adjacency graph (#issues in adjacency) is smaller than that in a connectivity graph (#issues in connection) because it also checks the route through potential connections. Thus, the ratio of #issues in adjacency to #issues in connection is equal or less than 1.

A smaller # issues in the adjacency graph than #issues in the current graph indicates some potential for error-correction by adding more doors. Thus, the ratio of #issues in adjacency graph to #issues in current connection graph could show the potential of fixing poor connections by adding doors. For instance, in the case of the courthouse 1, 112 issues are detected from the connectivity graph, and 107 issues are detected from the adjacency graph, indicating that five issues (= 112-107) have a valid route in the adjacency graph, and rest still remain an issue in the adjacency graph. In other words, we can say that 95.5% (107/112) of the issues are issues in both the adjacency and connection graphs, and 4.5% (5/112) of issues have the potential to be fixed.

This thesis defines the ratio of the number of issues in the adjacency graph to the number of issues in the connectivity graph as the "***real problem ratio***" because it shows the ratio of the real problems in both the adjacency and connectivity graphs. In addition, the value of (1- real problem ratio) is defined as the "***potential correction ratio***" because it shows the potential of correcting the issues from the connectivity graph.

Real problem ratio (RPR)

= the number of issues in the adjacency graph/the number of issues in the connectivity

graph

Potential correction ratio (PCR)

= (the number of issues in the connectivity graph - the number of issues in the adjacency

graph)/the number of issues in the connectivity graph

= 1 – Real problem ratio

Table 20 The validation results in terms of a real problem ratio and a potential correction ratio.

Table 20 The ratio of the issues from the connectivity to those from the adjacency graphs

Courthouse Name	**Modelling Date**	**Modelling Status**	**Summary of the Checking Results**		
			# issues	Real Problem Ratio x 100 (percentage)	Potential Correction Ratio x 100 (percentage)
Courthouse 1	2009/08/09	Final concept design	#issue : 112(107)	95.5 %	4.5%
Courthouse 2	2007/07/17	Design in development	#issue : 1(0)	0%	100%
	2007/08/24	Final concept design	#issue : 48 (43)	89.5%	10.5%
Courthouse 3	2009/11/12	Final concept design	#issue : 29(28)	96.5%	3.5%
Courthouse 4	2009/02/10	Final concept design	#issue : 43(36)	83.7%	16.3%
Courthouse 5	2008/05/27	Design in development	#issue : 31(25)	80.6%	19.4%
	2009/05/12	Design in development	#issue : 63(20)	31.7%	68.3%
	2009/07/24	Final concept design	#issue : 136(125)	91.2%	8.8%
	2009/08/07	Final concept design	#issue : 160(148)	92.5 %	7.5%
	2009/12/10	Final concept design	#issue : 173(159)	91.9%	8.1%

Table 20 shows that three early concept models—the courthouse 2 model 2007/07/17 model and the two courthouse 5 model -2008/05/27, 2009/05/12 models—have a higher Potential Correction Ratio (100% in the courthouse 2 model), (19.4% in the first courthouse 5- model), (68.3% in the second courthouse 5-model) than the other courthouses in the late development stage. The explanation for this finding is that the early stage of the design process affords more potential to create a valid route by adding doors than the late stage, which typically has fully-developed door-space connections. In the current testing, most of models in late concept stage exhibit a less than a 10% Potential Correction Ratio.

The Potential Correction Ratio exhibits a huge change depending on the design development stage. As considerable design development occurs from the second courthouse 5-model to the third one, the number of spaces increases from 357 to 1,238, and all the doors are defined for all spaces in the third model. Thus, the result of these changes during design development is a decreased Potential Correction Ratio from 68.3% to 8.8%.

However, the Potential Correction Ratio also varies among the courthouse designs. The courthouse 4 has a 16.3% Potential Correction Ratio even though it represents a late concept model. Even though the two courthouse 5-models, 2008/05/27 and 2009/05/12, are both early concept models, they exhibit huge differences in their Potential Correction Ratios because the 2009 model underwent many design changes (see Figure 59).

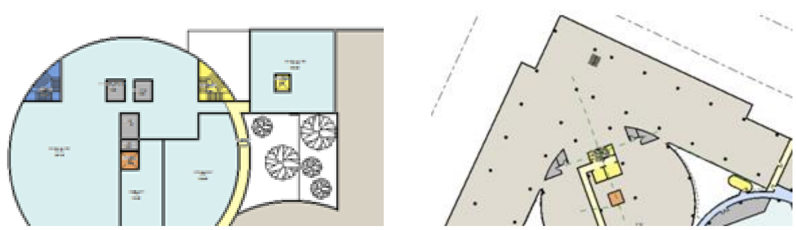


Figure 59 Courthouse 5-model 1 vs. courthouse 5-model 2

Another discussion issue that arose from the checking results is the ratio of the number of issues to the number of problem spaces. The number of issues caused by a problem space depends on the topological connection of spaces. In the case A connection in Figure 60, three problem spaces cause three issues for each terminal space, but in the case B connection, one problem space causes three issues. In this case, we could say that the issues are more concentrated in case B than in case A because in case B, the issues can be resolved by fixing one problem space, but in Case A, they can be resolved only by fixing of all three problem spaces.

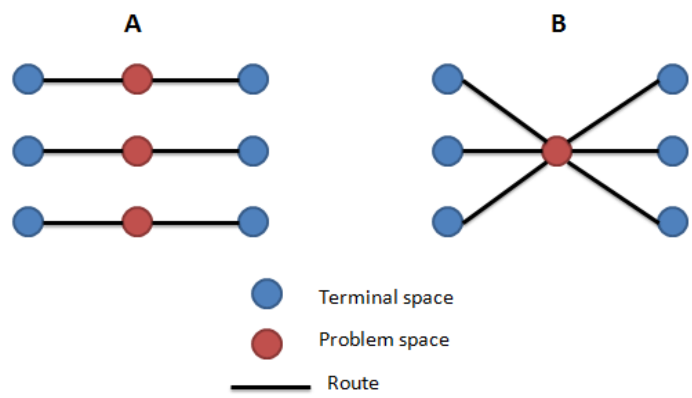


Figure 60 Two sample connections of spaces

The ratio of the number of issues to the number of problem spaces could be used to explain the concentration of issues. In the case of design A, the ratio is 1, but the ratio is 3 for design B. This thesis refers to this ratio as the Issue Concentration Ratio (ICR), and all of the ICRs of the models are calculated to determine the concentration of issues in all the real courthouse models (see Table 21).

Issue Concentration Ratio (ICR) = The number of issues/The number of problem spaces

The ICRs vary from 1.72 (courthouse 5-2008/05/27) to 5.25(courthouse 5-2009/05/12). The ICR for the courthouse 2 model (2007/07/17) is ignored because the number of issues is too small.

In the final concept design models, the ICRs vary from 2.63 to 5.25, suggesting that one problem space causes more than one issue on average.

To show the concentration of issues more clearly, this thesis calculates how many issues the top three problem spaces creates.

Table 21 The issue concentration ratio

Courthouse name	Modelling Date	Modelling status	Checking result Summary	
			# issues	Issue Concentration Ratio (#issue/ #pspace)
Courthouse 1	2009/08/09	Final concept design	#issue : 112 #pspace : 33	3.39
Courthouse 2	2007/07/17	Design in development	#issue : 1 #pspace: 4	0.25
	2007/08/24	Final concept design	#issue : 48 #pspace: 10	4.8
Courthouse 3	2009/11/12	Final concept design	#issue : 29 #pspace: 11	2.67
Courthouse 4	2009/02/10	Final concept design	#issue : 43 #pspace: 12	3.58
Courthouse 5	2008/05/27	Design in development	#issue : 31 #pspace: 18	1.72
	2009/05/12	Design in development	#issue : 63 #pspace: 12	5.25
	2009/07/24	Final concept design	#issue : 136 #pspace: 27	5.04
	2009/08/07	Final concept design	#issue : 160 #pspace: 33	4.84
	2009/12/10	Final concept design	#issue : 173 #pspace: 44	3.93

In the courthouse 1, the top three problem spaces—the corridor [3-900], the lobby [3-001], and the corridor [6-900]—cause 44, 44, and 16 issues, respectively. The issues caused by these problem spaces could be duplicated, so a unique sum of issues caused by the top three spaces is 64. It comprises 57.1% of the total number of issues.

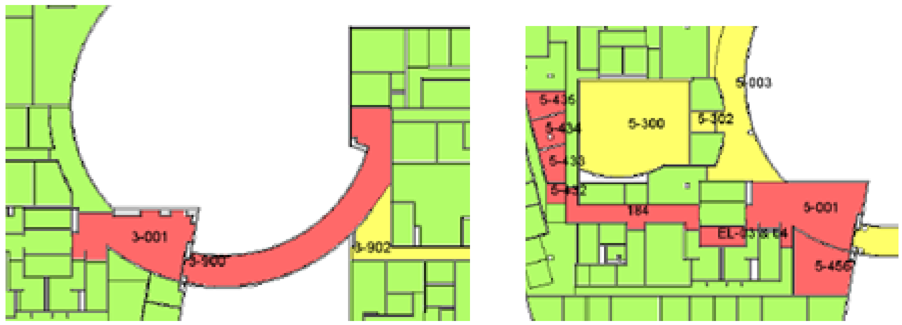


Figure 61 The top three problem spaces in the courthouse 1

The percentage of issues caused by the top three problem spaces among the total number of issues is calculated for all the courthouses. Table 21 shows the percentage of the top three problem spaces in each courthouse.

Table 22 The percentages of issues of the top three problem spaces of the total number of problems

Courthouse name	Modelling Date	Modelling status	Checking result Summary		
			# issues	Top three problem spaces	Percentage of top three problem spaces (#issue/ #Unique sum)
Courthouse 1	2009/08/09	Final concept design	#issue : 112	Corridor [3-900] causes 44 CORR. [184] causes 16 Deputy Clerks [2-370] causes 14 Sum: 74 Unique Sum: 64	57.1%
Courthouse 2	2007/07/17	Design in development	#issue : 1	Not calculated	Not calculated
	2007/08/24	Final concept design	#issue : 48	22(Storage[1238]) 18(Prisoner elevator[4217]) 17(Corrdior[1001]) Sum : 57 Unique sum: 41	85.4%
Courthouse 3	2009/11/12	Final concept design	#issue : 29	9 PUBLIC CORRIDOR [4162] RESTRICTED CORRIDOR [23] causes 8 RESTRICTED CORRIDOR [9] causes 8 Sum : 25 Unique sum : 24	82.7%

Courthouse name	Modelling Date	Modelling status	Checking result Summary		
			# issues	Top three problem spaces	Percentage of top three problem spaces (#issue/ #Unique sum)
Courthouse 4	2009/02/10	Final concept design	#issue : 43	Corridor [COR] causes 33 Corridor [COR] causes 19 Stair L1 [S1] causes 16 Sum : 68 Unique sum : 33	76.7%
Courthouse 5	2008/05/27	Design in development	#issue : 31	CORRIDOR [1321] causes 18 PUBLIC CORRIDOR [1261] causes 1 PUBLIC ELEVATOR [1389] causes 1 Sum: 20 Unique sum: 20	64.5%
	2009/05/12	Design in development	#issue : 63	PUBLIC CIRCULATION [4956] causes 20 PUBLIC CIRCULATION [3303] causes 11 VESTIBULE [1041] causes 5 Sum: 36 Unique sum: 23	36.5%
	2009/07/24	Final concept design	#issue : 136	PUBLIC CIRCULATION CORRIDOR [4956] causes 64 PUBLIC LOBBY [3303] causes 56 SHELVING /STORAGE AREA [1037] causes 5 Sum: 126 Unique sum: 70	51.4%
	2009/08/07	Final concept design	#issue : 160	PUBLIC CIRCULATION CORRIDOR [4956] causes 72 MAIN PUBLIC LOBBY [3303] causes 66 PRETRIAL/PROBATION RESTRICTED CIRCULATION [1046] causes 12 Sum: 150 Unique sum: 84	52.5%
	2009/12/10	Final concept design	#issue : 173	MAIN PUBLIC LOBBY [3303] causes 73 PUBLIC CIRCULATION CORRIDOR [4956] causes 55 PUBLIC CIRCULATION CORRIDOR [0000] causes 25 Sum: 153 Unique sum: 87	50.2%

The top three problem spaces of the courthouse 4 model has 68 problem spaces, but all issues in the second and third problem spaces overlap with the issues of the first problem space.

Table 23 The issues found in the Bakersfield model

Space Name	**Issues ID (Unique number for each issue)**
Corridor [COR1] causes 33	5,6,7,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,37,38,39,40,41
Corridor [COR2] causes 19	5,6,7,11,12,14,16,17,20,21,24,25,28,29,32,33,38,40,41
Stair L1 [S1] causes 16	6,7,11,12,16,17,20,21,24,25,28,29,32,33,40,41

Thus, the total number of unique issues is 33. This large number of issues occurred because of the area isolated by corridors 1 and 2 in terms of the security level. Figure 62 is a floor plan of the courthouse 4 color-coded according to the security level. Blue indicates a restricted zone, red a secure zone, green a public zone, and gray an unidentified zone. The figure shows that the restricted zone in the upper area is isolated from the other areas by a corridor (pointed out by arrow in the figure). The corridors are modelled just “cor,” so the system classifies the security level of the corridors as unclassified spaces. This isolation causes a number of security issues along the spaces around the corridors.

Figure 62 The floor plan of the courthouse 4

The courthouse 5-model has been developed continuously from models 2 to 5. Model 1 is the first model of the two courthouse 5-models, but huge design changes in most of the space connections took place during the design development process. Thus, model 1 is ignored from continuous design development of the courthouse 5-model.

Table 24 The top three problem spaces in the courthouse 5

Model Name	**Top Three Problem Spaces**
Courthouse 5*-1	Ignored.
Courthouse 5-2	PUBLIC CIRCULATION CORRIDOR [4956] causes 11 issues MAIN PUBLIC LOBBY [3303] causes 20 issues VESTIBULE [1041] causes 5 issues
Courthouse 5-3	PUBLIC CIRCULATION CORRIDOR [4956] causes 64 issues MAIN PUBLIC LOBBY [3303] causes 56 issues SHELVING /STORAGE AREA [1037] causes 5 issues
Courthouse 5-4	PUBLIC CIRCULATION CORRIDOR [4956] causes 72 issues MAIN PUBLIC LOBBY [3303] causes 66 issues PRETRIAL/PROBATION RESTRICTED CIRCULATION [1046] causes 12 issues
Courthouse 5-5	MAIN PUBLIC LOBBY [3303] causes 73 issues PUBLIC CIRCULATION CORRIDOR [4956] causes 55 issues PUBLIC CIRCULATION CORRIDOR [0000] causes 25 issues

**Model 1 is not counted in the study of continuous design development because of the numerous design changes that took place from Model 1 to Model 2.*

Table 24 shows the top three problem spaces in the courthouse 5-model from versions 2 to 5. The public circulation corridor and the main public lobby are continuously listed as problem spaces from versions 2 and 5, suggesting that design issues could remain until the final design if the design is developed without fixing the design issues found in the early stage of design.

9.2 Remaining issues

- **Well-formedness of the building model**

The purpose of the generic method is to check building models with diversity during the development stage. However, this method cannot be applied to any model in development because the model can have an almost infinite number of diverse forms according to the completeness and method of modelling. While performing preliminary research, this study identified several problems caused from the diversity of modelling. Some stairs were modelled by using slab as a landing between an upper-level stair and a lower-level stair instead of modelling the landing as a part of the stair [see Figure 63]. In some cases, the elevator space is modelled on every floor with less height than the distance from the floor to the ceiling instead of modelling the elevator as one space from the bottom floor and the top floor [see Figure 63].

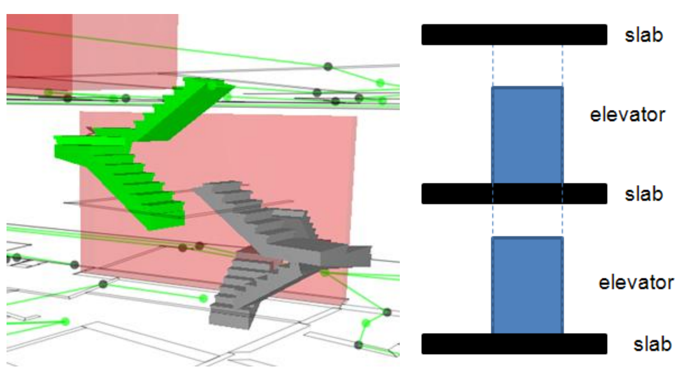


Figure 63 Diverse modeling of stairs and elevators

The integrated method should support a certain level of diversity of modelling so that a building modeller can have flexibility. Too rigorous requirements for checking can cause difficulty in modelling, and it is not practical to demand that a modeller follow too rigid requirements. However, modelling should maintain a certain level of well- formedness, and it is does not, the development of a computational algorithm applicable to any building model would not be feasible. This thesis also deal with the issue of well-formedness of a building model for checking occupant circulation by taking into account the flexibility of the modelling required to ensure that the method is practical.

- **Wrong space connection information**

The integrated checking module performs checking based on a graph generated from the IFC model. In particular, the connectivity graph is generated by using the space connection information in the IFC model. By the way, this thesis found several wrong space connections in the IFC model (see Figure 64). The wrong space connections were caused by poor modelling such as omitting a door, or by a "bug" in the BIM authoring tools. These inaccurate graph connections can lead to imprecise checking results. Thus, more study on the validation and the correction of the generated graph should be conducted.

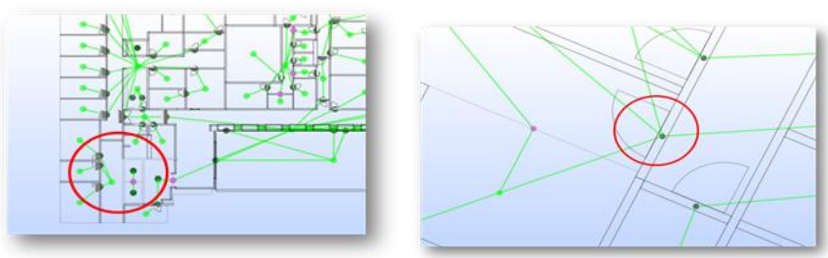


Figure 64 Wrong graph connections

- **Space Naming**

A space name is used as a key to identifying a space in the building model. Thus, the correct usage of a space name was a critical issue in identifying spaces in a building model for checking. However, numerous mistakes occur in the names of spaces caused by typos, the use of non-standard names, and the use of abbreviations. They cause problems in the identification of a space and inaccurate checking results. Thus, further study could refine the naming of spaces in a building model.

- **The pairing of start and target spaces**

A rule defines space names in a class level. If a rule is applied to a route in a model for checking, a start space and a target space corresponding to the start space should be selected. If two start and target spaces in the model are defined specifically enough to determine their relationship such as a bankruptcy court for judge XX and a judge's chamber for judge XX, then the route between them is the one that must checked when the rules between the courtroom and the judge's chamber are checked. However, such identifiers (e.g., the name or instance number) could vary even in the model, so their identification is very complicated. In most cases, models do not have these identifiers but instead they are denoted with class level names such as "courtroom" and "judge's chamber."

9.3 Conclusion

This thesis developed a generic method through experience from the development of set- and graph-based checking methods to check a model in varied stages of development. In an effort to develop a rule checking system that is more applicable to a real project, this research developed the more generic checking system based on a theoretical description of the checking module. By implementing the generic checking system, both the design developer and the final review can check a design in a shorter period of time.

The theoretical description and general class structure based on the theory are the first contributions to the body of knowledge relating to automatic rule checking during varied stages of design development. In addition, this research can serve as a basis for further research on the automatic checking of occupant rules for other type of buildings.

APPENDIX A

PARAMETRIC INTERPRETATION OF CIRCULATION RULES

This appendix shows the interpretation of circulation rules in U.S. Courts Design Guide by using parameters defined in this thesis. Followings are the definition of each parameter, and the table present how circulation rules are interpreted by using the parameters. You can find a circulation rule by using original (rule) id, which represent chapter, section and line number in U.S. Courts Design Guide 2007 version.

Definition of parameters

Original ID [X-XX-XXX(ID)]
X: Chapter, XX : Section, XXX: Line in 2007 U.S. Courts Design Guide, ID : Rule ID

Route Cardinality [AO | All]
AO : At least one, All : all

Start Space [Space Name]
Space Name : Name of space in Circulation Rules in U.S. Courts Design Guide

Target Space Cardinality [AO | All]
AO : At least one, All : all

Target Space[Space Name]
Space Name : Name of space in Circulation Rules in U.S. Courts Design Guide

Security Zone [PZ| RZ| SZ| n/a]
PZ: public zone, RZ : restricted zone, SZ : secure zone, n/a : not available

Space Usage [CR| PV|n/a]
CR : circulation, PV: Private, n/a : not available

Required Space [SpaceName | n/a]
Space Name : Required space name, n/a: not available

Vertical Access [not AW| AW| n/a]
not AW : not allowed, AW : allowed, n/a : not available

Maximum length [Distance(m) | n/a | RP]
Distance(m) : required maximum distance (m), n/a : not available, RP : Report Distance

Original ID	Route CD	Start space	Target space CD	target space	security zone	Usa-ge	required space	direct access	Vert-ical access	Maxi-mum-lengt-h
3-3-4(1)	all	Main entrance	all	Main PZ lobby	n/a	CR	Security screening area	Direct	not AW	n/a
3-3-18(2)	AO	Court of Appeals Clerk of Court	AO	PZ corridor	RZ	CR	n/a	n/a	n/a	RP
3-3-18(3)	AO	Bankruptcy Clerk of Court	AO	PZ corridor	RZ	CR	n/a	n/a	n/a	RP
3-3-18(4)	AO	District Clerk of Court	AO	District Judge Courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-18(5)	AO	Court of Appeals Clerk of Court	AO	Court of Appeals Panel courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-18(6)	AO	Court of Appeals Clerk of Court	AO	Court of Appeals En Banc courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-18(7)	AO	Bankruptcy Clerk of Court	AO	Bankruptcy Judge courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-18(8)	AO	District Clerk of Court	AO	District Judge Chambers	RZ	CR	n/a	n/a	AW	RP
3-3-18(9)	AO	Court of Appeals Clerk of Court	AO	Court of Appeals Judge Chambers	RZ	CR	n/a	n/a	AW	RP
3-3-18(10)	AO	Bankruptcy Clerk of Court	AO	Bankruptcy Judge Chambers	RZ	CR	n/a	n/a	AW	RP
3-3-18(11)	AO	District Clerk of Court	AO	PZ corridor	RZ	CR	n/a	n/a	n/a	RP
3-3-21(12)	AO	District Judge Chambers	AO	District Judge Courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-21(13)	AO	Senior District Judge Chambers	AO	Senior District Judge courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-21(14)	AO	Court of Appeals Judge Chambers	AO	Court of Appeals Panel courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-21(15)	AO	Court of Appeals Judge Chambers	AO	Court of Appeals En Banc courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-21(16)	AO	Bankruptcy Judge Chambers	AO	Bankruptcy Judge courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-21(17)	AO	Magistrate Judge Chambers	AO	Magistrate judge courtroom	RZ	CR	n/a	n/a	AW	RP
3-3-25(18)	AO	District Judge Chambers	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(19)	AO	Court of Appeals Judge Chambers	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(20)	AO	Bankruptcy Judge Chambers	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone), CR(Circulation), AW(Allowed), RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	security zone	Usa -ge	required space	direct access	Vert-ical access	Maxi-mum-lengt-h
3-3-25(21)	AO	Magistrate Judge Chambers	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(22)	AO	Senior District Judge Chambers	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(23)	AO	District law clerk office	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(24)	AO	senior district law clerk office	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(25)	AO	Magistrate law clerk office	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(26)	AO	Bankruptcy law clerk office	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(27)	AO	District Clerk of Court	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(28)	AO	Court of Appeals Clerk of Court	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-25(29)	AO	Bankruptcy Clerk of Court	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-37(30)	AO	Court of Appeals Judge Chambers	AO	Court of Appeals judge conference room	RZ	CR	n/a	n/a	AW	n/a
3-3-37(31)	AO	judge conference room	AO	judge robing room	RZ	CR	n/a	n/a	AW	n/a
3-3-37(32)	AO	Court of Appeals Panel courtroom	AO	Court of Appeals judge robing room	RZ	CR	n/a	n/a	AW	n/a
3-3-37(33)	AO	Court of Appeals En Banc courtroom	AO	Court of Appeals judge robing room	RZ	CR	n/a	n/a	AW	n/a
3-3-40(34)	AO	Circuit Executive's Office	AO	Court of Appeals Judge Chambers	RZ	CR	n/a	n/a	AW	n/a
3-3-40(35)	AO	Circuit Executive's Office	AO	Court of Appeals Clerk of Court	RZ	CR	n/a	n/a	AW	n/a
3-3-44(36)	AO	Staff Attorney Offices	AO	Court of Appeals Clerk of Court	RZ	CR	n/a	n/a	AW	n/a
3-3-44(37)	AO	Staff Attorney Offices	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	n/a
3-3-51(38)	AO	District Judge Courtroom	AO	Witness/Attorney Conference Room	PZ	CR	n/a	n/a	AW	RP
3-3-51(39)	AO	District Judge Courtroom	AO	Judges Conference / Robing Room	RZ	CR	n/a	n/a	n/a	n/a
3-3-51(40)	AO	District Judge Courtroom	AO	Trial Jury Room	RZ	CR	n/a	direct	n/a	RP

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone), CR(Circulation), AW(Allowed), RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	security zone	usage	required space	direct access	Vert-ical access	Maxi-mum-lengt-h
3-3-51(41)	all	District Judge Courtroom	AO	Courtroom holding cell	SZ	CR	n/a	n/a	n/a	n/a
3-4-4(42)	all	Main PZ lobby	all	Jury Assembly Room	PZ	CR	n/a	n/a	not AW	n/a
3-4-4(43)	AO	Jury Assembly Room	AO	District Clerk of Court	n/a	CR	n/a	n/a	not AW	RP
3-4-7(44)	AO	Grand Jury Hearing Room	AO	Office of the US Attorney	RZ	CR	n/a	n/a	AW	RP
3-4-7(45)	AO	Grand Jury Hearing Room	AO	Jury Assembly Room	RZ	CR	n/a	n/a	n/a	n/a
3-4-11(46)	AO	Security screening area	all	US Probation Office	PZ	CR	n/a	n/a	AW	n/a
3-4-14(47)	AO	Pretrial Services Office	AO	US Marshal Office	RZ	CR	n/a	n/a	AW	n/a
3-4-14(48)	AO	Pretrial Services Office	AO	Magistrate judge courtroom	RZ	CR	n/a	n/a	AW	n/a
3-3-51(49)	all	Vehicle Sallyport	AO	Central cell block	SZ	CR	n/a	n/a	AW	n/a
3-3-51(50)	all	Central cell block	AO	Courtroom holding cell	SZ	CR	USMS elevator	n/a	AW	n/a
3-3-51(51)	all	Courtroom holding cell	AO	District Judge Courtroom	SZ	CR	n/a	n/a	n/a	RP
3-3-51(52)	all	Courtroom holding cell	AO	Senior District Judge Courtroom	SZ	CR	n/a	n/a	n/a	RP
3-3-51(53)	all	Courtroom holding cell	AO	Magistrate judge courtroom	SZ	CR	n/a	n/a	n/a	RP
3-4-31(54)	AO	Bankruptcy Judge courtroom	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
3-4-31(55)	AO	Bankruptcy Judge courtroom	AO	RZ corridor	RZ	CR	n/a	n/a	AW	n/a
3-4-31(56)	AO	Bankruptcy Judge courtroom	AO	Witness/Attorney Conference Room	n/a	CR	n/a	n/a	n/a	RP
3-4-31(57)	AO	Bankruptcy Judge courtroom	AO	Judges Conference / Robing Room	RZ	CR				
3-4-36(58)	AO	Bankruptcy Administrator	AO	Bankruptcy Clerk of Court	RZ	CR	n/a	n/a	n/a	n/a
3-4-38(59)	AO	Bankruptcy Clerk of Court	AO	PZ corridor	PZ	CR	n/a	n/a	n/a	n/a
3-4-38(60)	AO	Bankruptcy Clerk of Court	AO	**Central Court Libraries**	RZ	CR	n/a	n/a	n/a	RP

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	security zone	usage	required space	direct access	Vert-ical access	Maxi-mum-length
3-4-38(61)	AO	Bankruptcy Clerk of Court	AO	District Clerk of Court	RZ	CR	n/a	n/a	n/a	RP
4-15-9(62)	AO	District Judge Courtroom	AO	PZ corridor	PZ	CR	n/a	n/a	n/a	n/a
4-15-9(63)	AO	District Judge Courtroom	AO	District law clerk office	RZ	CR	n/a	n/a	n/a	n/a
4-15-9(64)	AO	District Judge Courtroom	AO	District Court Reporter/Recorder Office	RZ	CR	n/a	n/a	n/a	n/a
4-15-9(65)	AO	District Judge Courtroom	AO	Trial Jury Room	RZ	CR	n/a	n/a	n/a	n/a
4-15-9(66)	AO	Senior District Judge Courtroom	AO	PZ corridor	PZ	CR	n/a	n/a	n/a	n/a
4-15-9(67)	AO	Senior District Judge Courtroom	AO	Senior District law clerk office	RZ	CR	n/a	n/a	n/a	n/a
4-15-9(68)	AO	Senior District Judge Courtroom	AO	Senior District Court Reporter/Recorder Office	RZ	CR	n/a	n/a	n/a	n/a
4-15-9(69)	AO	Senior District Judge Courtroom	AO	Trial Jury Room	RZ	CR	n/a	n/a	n/a	n/a
4-15-9(70)	AO	Senior District Judge Courtroom	AO	Courtroom holding cell	SZ	CR	n/a	n/a	n/a	n/a
4-24-9(71)	AO	Magistrate judge courtroom	AO	PZ corridor	PZ	CR	n/a	n/a	n/a	n/a
4-24-9(72)	AO	Magistrate judge courtroom	AO	Magistrate law clerk office	RZ	CR	n/a	n/a	n/a	n/a
4-24-9(73)	AO	Magistrate judge courtroom	AO	Magistrate Court Reporter/Recorder Office	RZ	CR	n/a	n/a	n/a	n/a
4-24-9(74)	AO	Magistrate judge courtroom	AO	Trial Jury Room	RZ	CR	n/a	n/a	n/a	n/a
4-24-9(75)	AO	Magistrate judge courtroom	AO	Courtroom holding cell	SZ	CR	n/a	n/a	n/a	n/a
4-31-9(76)	AO	Bankruptcy judge courtroom	AO	Bankruptcy law clerk office	RZ	CR	n/a	n/a	n/a	n/a
4-31-9(77)	AO	Bankruptcy judge courtroom	AO	Bankruptcy Court Reporter/Recorder Office	RZ	CR	n/a	n/a	n/a	n/a
4-31-9(78)	AO	Bankruptcy judge courtroom	AO	Trial Jury Room	RZ	CR	n/a	n/a	n/a	n/a
6-3-8(79)	AO	District Judge Chambers	AO	RZ corridor	RZ	CR	vestibule	n/a	not AW	n/a
6-3-8(80)	AO	Senior District Judge Chambers	AO	RZ corridor	RZ	CR	vestibule	n/a	not AW	n/a

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	security zone	Usa -ge	required space	direct access	Vert- ical access	Maxi- mum- lengt- h
6-3-8(81)	AO	Court of Appeals Judge Chambers	AO	RZ corridor	RZ	CR	vestibule	n/a	not AW	n/a
6-3-8(82)	AO	Bankruptcy Judge Chambers	AO	RZ corridor	RZ	CR	vestibule	n/a	not AW	n/a
6-3-8(83)	AO	Magistrate Judge Chambers	AO	RZ corridor	RZ	CR	vestibule	n/a	not AW	n/a
6-4-36(84)	AO	District Judge Chambers	AO	District Judge Toilet	n/a	CR	vestibule	n/a	n/a	RP
6-4-36(85)	AO	Senior District Judge Chambers	AO	Senior District Judge Toilet	RZ	CR	vestibule	n/a	n/a	RP
6-4-36(86)	AO	Court of Appeals Judge Chambers	AO	Court of Appeals Judge Toilet	RZ	CR	vestibule	n/a	n/a	RP
6-4-36(87)	AO	Bankruptcy Judge Chambers	AO	Bankruptcy Judge Toilet	RZ	CR	vestibule	n/a	n/a	RP
6-4-36(88)	AO	Magistrate Judge Chambers	AO	Magistrate Judge Toilet	RZ	CR	vestibule	n/a	n/a	RP
4-15-21(89)	AO	District Judge Courtroom	AO	Judges Conference / Robing Room	RZ	CR	n/a	n/a	not AW	n/a
4-15-21(90)	AO	Senior District Judge Courtroom	AO	Judges Conference / Robing Room	RZ	CR	n/a	n/a	n/a	n/a
4-15-21(91)	AO	Magistrate Judge Courtroom	AO	Judges Conference / Robing Room	RZ	CR	n/a	n/a	n/a	n/a
4-15-21(92)	AO	Bankruptcy Judge Courtroom	AO	Judges Conference / Robing Room	RZ	CR	n/a	n/a	n/a	n/a
4-15-26(93)	AO	District Judge Courtroom	AO	District Judge Chambers	RZ	CR	n/a	n/a	AW	RP
4-15-26(94)	AO	Senior District Judge Courtroom	AO	Senior District Judge Chambers	RZ	CR	n/a	n/a	AW	RP
4-24-23(95)	AO	Magistrate judge courtroom	AO	Magistrate Judge Chambers	RZ	CR	n/a	n/a	AW	RP
4-31-21(96)	AO	Bankruptcy Judge courtroom	AO	Bankruptcy Judge Chambers	RZ	CR	n/a	n/a	AW	RP
4-7-24(97)	AO	judge robing room	AO	Judge Toilet	RZ	CR	n/a	n/a	AW	RP
4-8-5(98)	AO	Attorney Workroom - Waiting Area	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-8-5(99)	AO	Court of Appeals Panel courtroom	AO	Attorney Workroom - Waiting Area	RZ	CR	n/a	n/a	AW	n/a
4-8-5(100)	AO	Court of Appeals En Banc courtroom	AO	Attorney Workroom - Waiting Area	RZ	CR	n/a	n/a	AW	n/a

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	security zone	Usa-ge	Requi-red space	direct ace-ss	Vert-ical access	Maxi-mum-lengt-h
4-16-8(101)	AO	District Judge Courtroom	AO	Attorney Workroom - Waiting Area	RZ	CR	n/a	n/a	AW	n/a
4-16-8(102)	AO	Senior District Judge Courtroom	AO	Attorney Workroom - Waiting Area	RZ	CR	n/a	n/a	AW	n/a
4-25-2(103)	AO	Magistrate judge courtroom	AO	Attorney Workroom - Waiting Area	RZ	CR	n/a	n/a	AW	n/a
4-32-2(104)	AO	Bankruptcy Judge courtroom	AO	Attorney Workroom - Waiting Area	RZ	CR	n/a	n/a	AW	n/a
4-16-14(105)	AO	District Court Reporter /Recorder Office	AO	RZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-16-14(106)	AO	Senior District Court Reporter/Recorder Office	AO	RZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-16-14(107)	AO	Bankruptcy Court Reporter/Recorder Office	AO	RZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-16-14(108)	AO	Magistrate Court Reporter/Recorder Office	AO	RZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-16-14(109)	AO	District Court Reporter/Recorder Office	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-16-14(110)	AO	Senior District Court Reporter/Recorder Office	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-16-14(111)	AO	Bankruptcy Court Reporter/Recorder Office	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
4-16-14(112)	AO	Magistrate Court Reporter/Recorder Office	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
11-5-12(113)	AO	Press / Media Room	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
11-5-12(114)	AO	Press / Media Room	AO	District Judge Courtroom	PZ	CR	n/a	n/a	not AW	RP
11-5-12(115)	AO	Press / Media Room	AO	Senior District Judge courtroom	PZ	CR	n/a	n/a	not AW	RP
11-5-12(116)	AO	Press / Media Room	AO	Court of Appeals Panel courtroom	PZ	CR	n/a	n/a	not AW	RP
11-5-12(117)	AO	Press / Media Room	AO	Court of Appeals En Banc courtroom	PZ	CR	n/a	n/a	not AW	RP
11-5-12(118)	AO	Press / Media Room	AO	Bankruptcy courtroom	PZ	CR	n/a	n/a	not AW	RP
11-5-12(119)	AO	Press / Media Room	AO	Magistrate courtroom	PZ	CR	n/a	n/a	not AW	RP
5-6-15(120)	All	Trial Jury Room	AO	PZ corridor	PZ	CR	Soundl-ock	n/a	AW	n/a

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	security zone	usage	required space	direct access	Vert-ical access	Maxi-mum-length
5-6-23(121)	AO	Grand Jury Hearing Room	AO	PZ corridor	RZ	CR	Entry Security Station	n/a	AW	n/a
7-6-2(122)	AO	Circuit Librarian	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	RP
7-6-2(123)	AO	Circuit Librarian	AO	Library CR Area	RZ	CR	n/a	n/a	AW	RP
7-6-2(124)	AO	Circuit Librarian	AO	Reference/Card Catalog Area	RZ	CR	n/a	n/a	AW	RP
10-7-4(125)	AO	Circuit Executive's Office	AO	PZ corridor	RZ	CR	n/a	n/a	AW	n/a
10-7-14(126)	AO	Senior Staff Attorney	AO	RZ corridor	RZ	CR	n/a	n/a	AW	n/a
10-7-14(127)	AO	Senior Staff Attorney	AO	Library Entry/Lobby	RZ	CR	n/a	n/a	AW	RP
10-7-14(128)	AO	Senior Staff Attorney	AO	Court of Appeals Clerk of Court	RZ	CR	n/a	n/a	AW	RP
10-7-17(129)	AO	Senior Conference Attorney Office	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
10-7-11(130)	AO	Office of the District Court Executive	AO	RZ corridor	RZ	CR	n/a	n/a	AW	n/a
10-7-11(131)	AO	Office of the District Court Executive	AO	District Judge Chambers	RZ	CR	n/a	n/a	AW	n/a
10-7-11(132)	AO	Office of the District Court Executive	AO	PZ corridor	RZ	CR	n/a	n/a	AW	RP
10-7-7(133)	AO	Office of the Bankruptcy Appellate Panel Clerk	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
10-7-7(134)	AO	Office of the Bankruptcy Appellate Panel Clerk	AO	Bulk storage area	RZ	CR	n/a	n/a	AW	n/a
9-7-2(135)	AO	US Probation Office	AO	Reception / Waiting Area	PZ	CR	n/a	n/a	AW	n/a
9-7-2(136)	AO	Pretrial Services Office	AO	Reception / Waiting Area	PZ	CR	n/a	n/a	AW	n/a
9-7-2(137)	AO	US Probation Office	AO	RZ corridor	RZ	CR	n/a	n/a	AW	n/a
9-7-2(138)	AO	Pretrial Services Office	AO	RZ corridor	RZ	CR	n/a	n/a	AW	n/a
9-2-10(139)	All	Urinalysis Testing Lab	AO	RZ corridor	RZ	CR	n/a	n/a	AW	n/a
9-2-10(140)	AO	Urinalysis Testing Lab	AO	Urinalysis Testing Toilets	RZ	CR	n/a	n/a	not AW	RP

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	security zone	usage	required space	direct access	Vert-ical access	Maxi-mum-lengt-h
9-2-10(141)	AO	Urinalysis Testing Lab	AO	Urinalysis Supplies Storage	RZ	CR	n/a	n/a	not AW	RP
9-3-43(142)	AO	Urinalysis Testing Toilets	AO	Reception / Waiting Area	RZ	CR	n/a	direct	not AW	n/a
9-4-1(143)	AO	Urinalysis Testing Toilets	AO	Urinalysis Supplies Storage	RZ	CR	n/a	direct	not AW	n/a
10-7-19(144)	AO	Bankruptcy Administrator	AO	PZ corridor	PZ	CR	n/a	n/a	AW	n/a
8-2-19(145)	AO	District Clerk of Court	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	RP
8-2-19(146)	AO	District Clerk of Court	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	RP
8-2-19(147)	AO	Bankruptcy Clerk of Court	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	RP
8-2-19(148)	AO	Bankruptcy Clerk of Court	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	RP
8-2-19(149)	AO	Court of Appeals Clerk of Court	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	RP
8-2-19(150)	AO	Court of Appeals Clerk of Court	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	RP
10-2-2(151)	AO	Senior Staff Attorney	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	RP
10-2-2(152)	AO	Senior Staff Attorney	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	RP
9-2-29(153)	AO	US Probation Office	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	AW	RP
9-2-29(154)	AO	US Probation Office	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	RP
9-2-29(155)	AO	Pretrial Services Office	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	RP
9-2-29(156)	AO	Pretrial Services Office	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	RP
10-2-2(157)	AO	Office of the Federal Defender	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	RP
10-2-2(158)	AO	Office of the Federal Defender	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	RP
10-2-2(159)	AO	Circuit Executive's Office	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	n/a
10-2-2(160)	AO	Circuit Executive's Office	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	n/a

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	Secu -rity zone	Usa -ge	required space	direct access	Vert-ical access	Maxi-mum-lengt-h
10-2-2(161)	AO	Office of the District Court Executive	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	n/a
10-2-2(162)	AO	Office of the District Court Executive	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	n/a
10-2-2(163)	AO	Senior Conference Attorney Office	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	n/a
10-2-2(164)	AO	Bankruptcy Appellate Panel Clerk	AO	Staff Lounge/Break Room	RZ	CR	n/a	n/a	not AW	n/a
10-2-2(165)	AO	Senior Conference Attorney Office	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	n/a
10-2-2(166)	AO	Bankruptcy Appellate Panel Clerk	AO	Staff Toilets	RZ	CR	n/a	n/a	not AW	n/a
16-4-39(167)	AO	District Judge Chambers	AO	Judges Parking Area	RZ	CR	n/a	n/a	n/a	n/a
16-4-39(168)	AO	Court of Appeals Judge Chambers	AO	Judges Parking Area	RZ	CR	n/a	n/a	n/a	n/a
16-4-39(169)	AO	Bankruptcy Judge Chambers	AO	Judges Parking Area	RZ	CR	n/a	n/a	n/a	n/a
16-4-39(170)	AO	Magistrate Judge Chambers	AO	Judges Parking Area	RZ	CR	n/a	n/a	n/a	n/a
16-4-39(171)	AO	Senior District Judge Chambers	AO	Judges Parking Area	RZ	CR	n/a	n/a	n/a	n/a
11-5-17(172)	AO	Loading dock	all	Maintenance shop, bulk building services	RZ	CR	n/a	n/a	AW	RP
11-5-17(173)	AO	Loading dock	all	Bulk storage area	RZ	CR	n/a	n/a	AW	RP
11-5-17(174)	AO	Loading dock	all	Maintenance shop, equipment and furniture	RZ	CR	n/a	n/a	AW	RP
11-5-17(175)	AO	Loading dock	all	Maintenance shop, general surplus material	RZ	CR	n/a	n/a	AW	RP
11-5-17(176)	AO	Loading dock	all	Archive storage	RZ	CR	n/a	n/a	AW	RP
11-5-17(177)	AO	service elevator	all	Maintenance shop, bulk building services	RZ	CR	n/a	n/a	AW	RP
11-5-17(178)	AO	service elevator	all	Bulk storage area	RZ	CR	n/a	n/a	AW	RP
11-5-17(179)	AO	service elevator	all	Maintenance shop, equipment and furniture	RZ	CR	n/a	n/a	AW	RP
11-5-17(180)	AO	service elevator	all	Maintenance shop, general surplus material	RZ	CR	n/a	n/a	AW	RP

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

Original ID	Route CD	Start space	Target space CD	target space	Security zone	Usage	required space	direct access	Vert-ical access	Maxi-mum-lengt-h
11-5-17(181)	AO	service elevator	all	Archive storage	RZ	CR	n/a	n/a	AW	RP
16-6-12(182)	AO	Jury Assembly Room	all	Trial Jury Room	RZ	CR	n/a	n/a	AW	n/a
16-6-12(183)	AO	Jurors' Dining room	all	Trial Jury Room	RZ	CR	service elevator	n/a	AW	n/a

Notation : Route CD(Cardinality) , AO (At least one), PZ(Public Zone), RZ(Restricted Zone), SZ(Secure zone, RP(Report)

REFERENCES

[1] Vassileva, S.,(2000). An approach of constructing an integrated client/sever framework for operative checking of building code. Construction informatics digital library: p. Paper w78-2000-996.

[2] Xu, R., Solihin, W., Huang, Z.,(2004). Code Checking and Visualization of an Architecture Design, Visualization. IEEE Journal on Selected Areas in Communications, 10(15): p. 10.

[3] Ding, L., Drogemuller, R., Jupp, J., Rosenman, M.A., and Gero, J. S.,(2004). Automated code checking, in CRC for Construction Innovation, Clients Driving Innovation International Conference, pp. 25–27 .

[4] Han, C. S., Law, K. H.; Latombe, J. C., Kunz, J-C., (2002). A performance-based approach to wheelchair accessible route analysis.Advanced Engineering Informatics, 16(1): pp. 53-71.

[5] Han, C. S., Kunz, J. C., Law, K. H.,(1998). Hybrid prescriptive-performance-based approach to automated building code checking, International Computing Congress, ASCE, pp. 537-548.

[6] Farinha, F.,(2005). Code checking automation in building design: New trends for cognition Santos, in the 2005 ASCE International Conference on Computing in Civil Engineering. pp. 143-150.

[7] Nguyen, T.-H., (2005). Integrating building code compliance checking into a 3D CAD system, in the 2005 ASCE International Conference on Computing in Civil Engineering. 2005. pp. 1009-1020.

[8] Nguyen, T.-H., Oloufa, A. A.,(2002). Spatial information: Classification and applications in building design. Computer-Aided Civil and Infrastructure Engineering, 17(4): pp. 246-255.

[9] Nguyen, T. H., Oloufa, A.A., and Nassar, K.,(2005). Algorithms for automated deduction of topological information. Automation in Construction, 14(1): pp. 59-70.

[10] Yang, Q. Z. and Xu, X. J., (2004). Design knowledge modeling and software implementation for building code compliance checking. Building and Environment, 39(6): pp. 689-698.

[11] Yang, Q. Z., Lu, W. F.,(2004). Development of a J2EE web application for step-based design conformance checking, in Proceedings of the ASME Design

Engineering Technical Conference and Computers and Information in Engineering, pp. 1011-1019 .

[12] Kincho, J. K., Han, C.S., Kunz, J. C., Law, Kincho K. H., (2000). A distributed object component-based approach to large-scale engineering systems and an example component using motion planning techniques for disabled access usability analysis. Computing in Civil and Building Engineering, 1: p. 627-634.

[13] Tang, H. N., Bedard, C., Kinh, H. H.,(1996). Automated code compliance checking for building inspection, in International Computing Congress, ASCE, , pp. 1020-1026.

[14] Delis, E. A. and Delis, A., (1995). Automatic Fire-Code Checking Using Expert-System Technology. Journal of Computing in Civil Engineering, 9(2): pp. 141-156.

[15] Lopez, L. A., Elam, S.; Reed, K.,(1989). Software concept for checking engineering designs for conformance with codes and standards. Engineering with Computers, 5(2): pp. 63-78.

[16] Satti, H. M., Krawczyk, R. J., (2004). Issues of integrating building code in cad, 1st ASCAAD International Conference, e-Design in Architecture KFUPM, pp89-107.

[17] Fatt, C.T.,(2005). The Pillar of IFC implementation in Singapore, Building & Construction Authority Singapore. Available: http://www.iai.no/2005_buildingSMART_oslo/Session%2001/eSubmission_eplancheck_Singapore_Case.pdf

[18] Shirley, E.,(1974). Circulation analysis and lift placing in hospitals. Computer-Aided Design, 6(4): pp. 206-21.

[19] Chiu, C.-Y., (2002). Michael Walton, C., Integrated simulation method to evaluate the impact of new large aircraft on passenger flows at airport terminals. Transportation Research Record, (1788): pp. 83-92.

[20] Polillo, R. R.,(2002). Secure flow a totally integrated transportation security system. In IEEE Annual International Carnahan Conference on Security Technology.

[21] Yoshida, T.T.K.,(2008). Improvement of Pedestrian Shop-around BehaviourAgent Model: Design and Implementation of “Erratic Visit” Behaviour Model. Design & Decision Support Systems in Architecture and Urban Planning, University of Technology Eindhoven, published on CD.

[22] Sema E. and Palaniappa K. (2000), In-Patient flow analysis using PROMODEL simulation package, FREC SPOO-O2, University of Delaware.

[23] Karunakaran, A.,(2005). Organization of Pedestrian Movements: an Agent-Based Approach, in CAADRIA: 10th International Conference on Computer Aided Architectural Design Research in Asia, New Delhi (India), vol. 1, pp. 305-313.

[24] Zhu, W. and Timmermans, H.J.P. (2007). Approach to incorporating principles of bounded rationality into models of individual decision making." In: Proceedings RARSS Conference, San Francisco., USA, CD-Rom, 19 pp.

[25] Joseph L. S., James T. B., and James T. Brokaw, P.E., (2008). Agent based simulation of human movements during emergency evacuation of facilities, ASCE Conference Proceeding , pp. 1-10

[26] Van der Waerden, P., Borgers, A. and Timmerinans, H.,(1996). Route related data of shopping centre visitors and geographical information systems. In Third Design and Decision Support Systems in Architecture and Urban Planning - Part two: Urban Planning Proceedings. Spa, Belgium.

[27] Aloys, W.J., Borgers, I.M.E.Smeet., Kemperman, A.D.A.M, Kemperman and H.J.P. Timmermans.,(2006). Simulation of Micro Pedestrian Behaviour in Shopping Streets.In Progress in Design & Decision Support Systems in Architecture and Urban Planning. Eindhoven: Eindhoven University of Technology, pp101-116.

[28] Van der Waerden, P., Van de Voort, R. and Da Silva, A.N.R.,(2002). Studying Pedestrian Movements in Central Shopping and Business Areas with a Dedicated Geographical Information System. In Sixth Design and Decision Support Systems in Architecture and Urban Planning - Part two: Urban Planning Proceedings. Avegoor, Netherland, pp251-261.

[29] Koutamanis, A., (1995). Multilevel Analysis of Fire Escape Routes in a Virtual Environment.In Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures, Singapore, pp. 331-342.

[30] Nassar, K. , Nguyen T-H, (2005). Using Building Topological Information to Check for Means of Egress Building Code Compliance.ITCon, 2005. 10: pp. 219-227.

[31] Andreatta, G., Brunetta, L. Righi, L.,(2007). Evaluating terminal management performances using SLAM: The case of Athens International Airport. Computers and Operations Research, 34(6): pp. 1532-1550.

[32] Kunigahalli, R. Veeramani, D.,(1995). Extracting Topological Relationships from Wire-Frame CAD Model Journal of Computing in Civil Engineering, 9(1): pp. 29-42.

[33] Kannala, M.,(2005). Escape Route Analysis Based on Building Information Models: Design and Implementation, in MS thesis, Department of Computer Science and Engineering, Helsinki University of Technology.

[34] Rabin, S., (2003). AI Game programming wisdom2, Publisher : Charles River Media

[35] Grason, J.,(1971). An approach to computerized space planning using graph theory, Proceedings of the Design Automation Workshop, June 28-30, Atlantis City, N.J., New York, pp. 170-179.

[36] Krejcirik, M.,(1969). Computer-Aided Plant Layout. Computer-Aided Design,volume pp. 7-19.

[37] Levin, P.H.,(1964). Use of Graphs to Decide the Optimum Layout of Buildings.Architects' Journal, pp. 809 – 815.

[38] Solihin, W.,(2004). Lessons learned from experience of code-checking implementation in Singapore, novaCITYNET Pte. Ltd., Available: http://www.buildingsmart.com/files/u1/ned_20from_20experience_20of_20code_checking.pdf

[39] Eastman, C., Jung, Y-S., Lee, J-M., Lee, J. K.,(2007). GSA-BIM Enabled Design Guide. Project report, Georgia Institute of Technology: Atlanta.

[40] Lee, J-M., Eastman, C. , Jung, Y-S., Lee, J. K.,(2007). Parametric representation of pedestrian circulation rules in U.S. Courts Design Guide, Project report, Georgia Institute of Technology: Atlanta.

[41] Muther, R.,(1973). Systematic Layout Planning, Cahners Books, Boston

[42] Foulds, L.R.,(1983). Techniques for facilities layout: Deciding which pairs of facilities should be adjacent, Management Science, Vol. 29, pp. 1414-1426.

[43] Okazaki, S. and Matsushita, S.,(1993). A study of simulation model for pedestrian movement with evacuation and queuing, Proceeding of the International Conference on Engineering for Crowd Safety, pp.271-280.

[44] U.S. General Service Administration, (2010). Design and Construction Delivery Process, Available:

http://www.gsa.gov/Portal/gsa/ep/contentView.do?contentType=GSA_BASIC&contentId=10366

[45] Dijkstra, E. W., (1959). A note on two problems in connexion with graphs.NumerischeMathematik 1: 269–271.

[46] Asano, M., Kuwahara, M., Sumalee, A., and Chung, E., (2006) Pedestrian simulation considering stochastic route choice and multi directional flow. 2nd International Symposium of Transport Simulation (CD), EPFL, Lausanne. Switzerland, pp.47-59

[47] Lee, J.K., Eastman, C.M., Lee, J.M., Jeong, Y.S., Kannala, M., Accessible distance measurement using the GT Metric Graph (2008). CIB conference, Georgia Tech, Atlanta, USA. pp.142-149